AN ARGUS SPECIALIST PUBLICATION

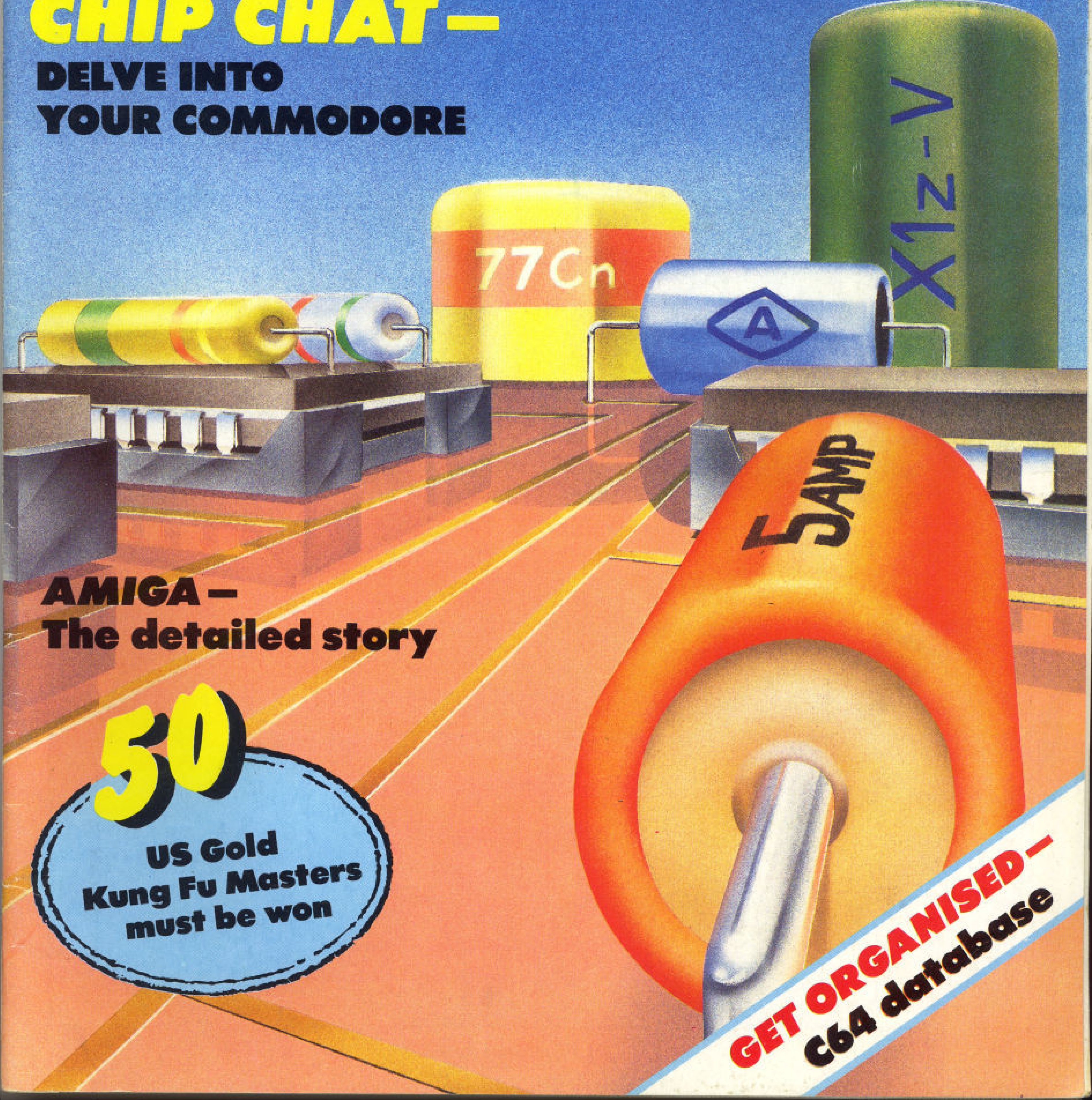
JUNE 1986

£1.00

Your COMMODORE

YOUR BEST INDEPENDENT COMMODORE MAGAZINE

CHIP CHAT —
DELVE INTO YOUR COMMODORE



AMIGA — The detailed story

50 US Gold Kung Fu Masters must be won

GET ORGANISED — C64 database



WINTER EVENTS
C16/PLUS 4
Compete in the six thrilling events - Biathlon, Bobsled, Ski-Jump, Slalom, Down Hill and Speed Skating. The events start with opening ceremony accompanied by thrilling music. One to four players can select the country they wish to represent. The events with some of the most stunning graphics seen on C16/PLUS 4 provide a perfect back drop as the chase for medals progresses. TURBO LOAD ensures that you don't get a chance to cool down between the events.
A QUALITY GAME
FOR C16 AND PLUS 4 OWNERS
£7.95
ANCO
Trade Enquiries:
C16-PLUS 4 CENTRE, ANCO SOFTWARE,
85, TILE KILN LANE, BEXLEY, KENT, U.K.
Telephone: 0322 92513/8
Mail Order: Payment by Cheque, P.O., VISA or ACCESS
24 Hour Credit Card Service: 0322 522631

JUNE 1986

VOLUME 2
NUMBER 9

Editor:
Stuart Cooke

Assistant Editor:
Marie Curry

Advertisement Manager
John McGarry

Advertisement Copy Control:
Laura Champion

Group Editor:
Dave Bradshaw

Group Managing Editor:
Wendy Palmer

Managing Director:
Peter Welham

Origination:
Ebony Typesetting

Design:
Argus Design

Editorial & Advertisement Office
No 1 Golden Square,
London W1R 3AB.
Telephone: 01-437 0626
Telex: 8811896

Your Commodore incorporating Your 64 is a monthly magazine appearing on the first Friday of each month.

Distribution by: Argus Press Sales & Distribution Ltd. 12-18 Paul Street, London EC2A 4JS. Printed by: Alabaster Passmore & Sons Ltd, Tovil, Maidstone, Kent.

Subscription rates upon application to Your Commodore Subscriptions Department, Infonet Ltd, Times House, 179 The Marlowes, Hemel Hempstead, Herts. HP1 1BB.

CONTENTS

FEATURES

SERIES

REGULARS

COMPETITION

DATA STATEMENTS

Soft in the Head

SHOPPING FOR SOFTWARE? LOOK NO further than these pages. This month it seems as if there's never been a wider choice of program available and you have to congratulate the software houses in the ingenuity of their staff.

Ariolasoft Info

Ariolasoft is making headlines and movies this month. Did you ever think that you could play at being Steven Spielberg with the aid of your C64? Movie Maker allows you to do just that. You can create your own animated pictures with the aid of this program. You probably won't make as much money as the great directors but you should have some fun.

Claims for the program include: mixing text and graphics, creating tracks of up to 30 frame each, and combining six of these in a single animated sequence.

There are also hundreds of built in pictures, ready for use plus the power to create your own. There are numerous animated greetings cards and a music and sound library. You can videotape your efforts and bore your friends with them – if you feel that way inclined.

It's on twin disks and costs £16.95.

Two other newies from this Covent Garden based house – they've just moved – are Starship Andromeda and Panzadrome.

The latter is a futuristic war game set on a robot inhabited island. The Robots are tanks and hostile, of course, and you must build a better one in order to defeat them. Look out for mines and mortars.

Panzadrome costs £8.95 and is a product of the programming and design skills of the Ramjam Corporation.

Starship Andromeda sees you battling the evil tyrant Alana, who wears a crystal around her neck which just happens to be the security key to the great computer which controls the galaxy. A tough situation – especially since the only thing that will free the crystal is a Proton lance. Get the lance and get the crystal. Easy, huh?

The program includes 12 arcade/strategy games and 10,000 locations. It's also £9.95.

Not content with these launches, Ariolasoft is entering the Amiga software market with a collection of Electronic Arts' utilities.

Deluxepaint is an incredible graphics program which allows the beginner to explore the Amiga's capabilities whilst giving the more experienced user an excellent tool. Features include tools to create lines, curves, shapes and textures. There's also split screen zoom enlarging and colour cycling to create animation.

DeluxePrint is a sister program with 15 formats for easy customisation of print outs – in colour or black and white. The formats include greetings cards, business cards, stationary, calendars, letterheads, posters and many more.



▲ Amiga range from Ariolasoft

▼ Ariolasoft, Electronic Arts and the Amiga



DeluxeVideo brings another dimension to computing. Make video slide shows, animation, business presentations. Find out about wipes, fades, and dissolves. Also compression of images, control of the foreground and background and much more.

Ariolasoft's Ashley Gray said: "The Deluxe series of Amiga programs are the most powerful, integrated, creative tools ever released for a personal computer."

Maybe you should try them when you've saved up for your Amiga?

Activision — Out of Time?

WANTED – DEAD OR ALIVE! SAM Harlow, that's you, a marked man. Your past has caught up with you, punk. They're out to get your – but you don't know who they are. Check your files, Sam, you need to prevent your own murder, and it's one of those bums who you've put away behind bars during your auspicious career.

Interested? Activision's latest adventure – Borrowed Time – puts you in the position of both hunter and prey. There are 20 serious suspects who are out of your blood. Track the murderer down before it's too late.

Perils including getting shot at, burned, anaesthetised, beaten up, strung up, sentenced to life imprisonment, bopped on the head, ripped to pieces by mad dogs etc.

If you've got £14.99 and want to die horribly then you know how to spend it.

And the Rest

IF YOU'RE AN AVID WATCHER OF NOEL Edmond's ridiculous Late Late Breakfast Show, then you'll know the background to Mr Puniverse. Now Mastertronic has jumped on the bandwagon yet again to bring you Big Mac – 2 which features the character immortalised in this hilarious TV spoof. It's on the C-16 and is in the £1.99 range of course.

Halley's Comet fever has now hit the software industry with a release from Firebird to mark the approach of the space probe Giotto to the heart of the comet.

It's one of those games in which, once again, you play the part of an intrepid spaceman coming to the aird of the entire human race. This time the danger involves germ bags from the comet which are threatening the health of the world.

The game has three phases starting with the launch, followed by controlling the ship's computer in its flight to the comet (the spaceman is in suspended animation at this point), and finally the destruction of the germbags so that they won't reproduce and infect the globe. £7.95 on the C64, and available now.

Rino Software is taking the bull by the horns in its decision to call a spade a spade – or more accurately a shoot 'em up a shoot 'em up.



Gone are the days when zappy games had zappy names. Rino's new release is for the C64, priced at £2.99 and is unabashedly entitled Shoot 'Em Up.

Generally Speaking

IN THIS HIGHLY TECHNOLOGICAL BUSiness, the Japanese are still the people who are often first with the newest ideas. Now Cumana has brought another Japanese invention to this country in the form of the Astron IC card.

Physically, the card resembles a credit card but contains integrated circuitry which enables it to be used as a solid state memory device. It has a 38 pin connector which plugs into a cheap (about £2) socket and does not need a magnetic or laser reader.

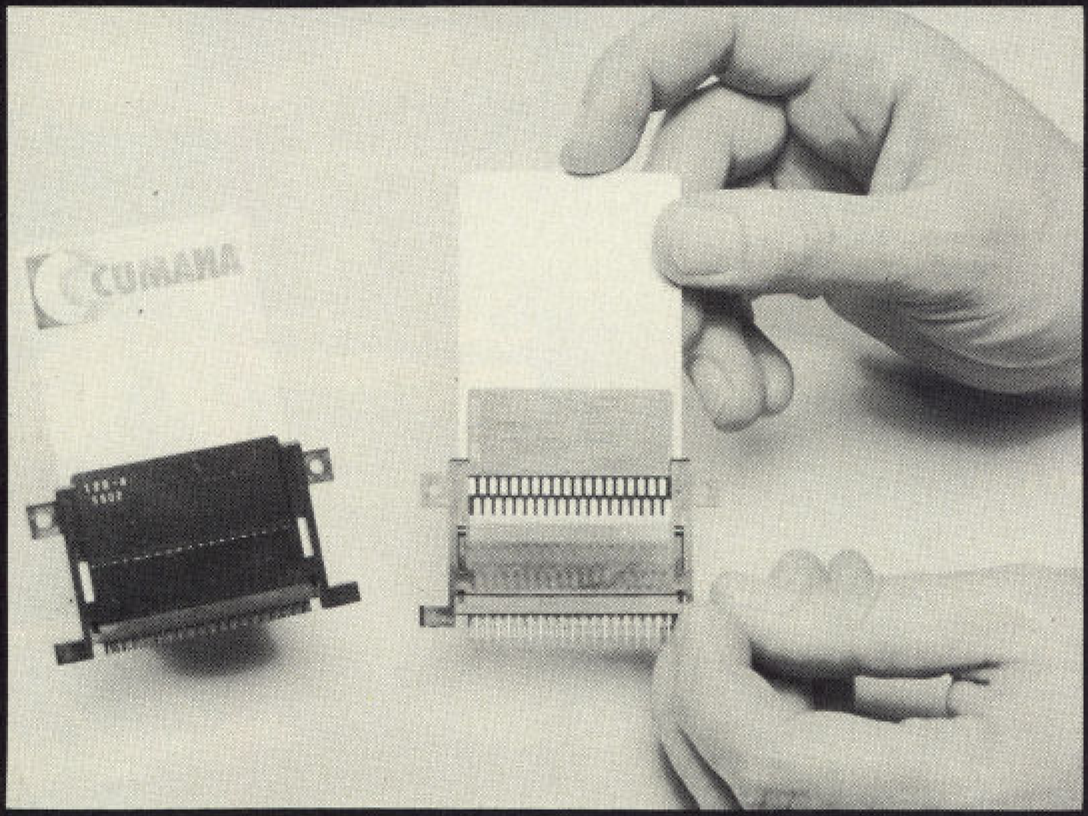
There are four different types of card: ROM, Masked ROM, EPROM and RAM.

Cumana claims that they are totally reliable and durable and are unaffected by electromagnetic/electrostatic conditions, extreme temperatures, humidity and minor scratches. It is also an added protection against software piracy.

Adaptors for the cards will be available for the C64 and C128.

Cumana has proposed a host of applications for the new card: Memory for telecommunication devices; identification for security keys, bank accounts, computer input and cashless shopping; speech for vending machines, language labs, measuring and sensing warning devices; instructions for robots and other automated equipment; personal identity; applications and ROM software; portable memory for hand held micros, programmable printers, process control, alarm systems. And this is only the tip of the iceberg. Look out for them.

The Astron Integrated Circuit Card





Mike Mahoney and Tony Crowther

After an absence of two years, Games programmer Tony Crowther has returned to Alligata software and as you can see from the picture he seems to be having quite a good time there already.

Because of this new arrangement Alligata is promising some exciting releases in the near future.

If you've bought an A'n'F game recently or in the past then a new service is being offered to you by the company to help you out with any problems which you may have run into.

A new mail order and queries line has been established which will provide a 24 hour service. The number is Rochdale (0706) 341111. There's also been an announcement from A'n'F that a new arcade adventure is in the pipeline at the moment. We'll bring you more news on that as soon as we have it.

A company called Mupados has come up with an original idea. It has launched a software/audio cassette. It combines theme tunes in 'full, fantastic stereo' dance mix from some popular games – namely Rambo, Neverending Story, Ghostbusters, Crazy Comets and Hypersports – with a database called Softwhere? which should help you organise a reference system for your programs, games, or record collection.

The idea behind it is apparently to combine the purely functional with pure entertainment. Whether the punters will think this worthwhile is another question but at £4.99 from W H Smiths it may be worth a shot.

On Line

MICRONET USERS HAVE RALLIED round and forked out to make the lives of some disabled young people a lot easier.

A user called Jon Bye, who became non-vocal as the result of an accident several years ago, let Micronet know, via mailbox, that communications services had made an incredible difference to his life.

His comments created a lot of interest amongst other users. John lives at Hatchford Park School in Guildford and an appeal put up on Micronet to raise £300 for 10 other non-vocal residents was greeted with a fantastic response.

Not only did members donate over £350 in 24 hours, but also Keith Rose of Modem House offered all the required modems free.

Hard Lines

COMMODORE 128 OWNERS WHO have had trouble finding a suitable monitor for their computer (other than Commodore's own product) could find that Cascade has come to their aid.

The RGBI output of the 128 is incompatible with the majority of monitors on the market. Commodore's 1901 monitor is expensive and people who already own a Microvitec probably feel that this is a high price to pay.

Cascade has come up with an interface which provides a full RGB 80 column display via the RGB TTI input. Microvitec 1431 dual mode monitor can now provide full 40 column composite video and 80 column RGB with simple switching between modes, thus allowing full use of the 128's three operating modes.

At £19.95 it could be a worthwhile investment.

Musical computer owners will be tempted by Commodore's new Complete Music System. It costs £330 including a C64 and the Music Expansion system priced at £149.99 for those who already own a C64 or C128, contains everything needed to produce music which is normally only possible on expensive synthesisers. Or so Commodore claims.

Launched at the Ideal Home Exhibition the products should be in your local shop now.

Rick Wakeman tackles Commodore's Music System





THE WAY OF THE TIGER
Enter the world of Avenger, a Ninja warrior of unparalleled skills and deadly powers, as he battles the forces of evil in defence of his faith and protection of the weak.
Be calm and stay silent as the outstanding animation and unrivalled combat routines take you to levels of action you'd never have thought possible.
Experience the stunning effects of triple scrolling action as you master the techniques of Hand to Hand combat, Pole Fighting and the skills of the Samuri Sword.
And when you believe you've succeeded in overcoming all the odds, the next in this thrilling series of adventures will beckon you forward to a further challenge of death.
Cassette £9.95
Disc £14.95
MSX 64K Spectrum 48K Amstrad CBM 64/128
Gremlin Graphics Software Limited, Alpha House, 10 Carver Street, Sheffield S1 4FS. Tel: 0742-753423
TRIPLE-SCROLL
TRIPLE-SCROLL
Screenshots from Spectrum 48K

COMPETITION

Martial arts without pain can be yours if you enter our US Gold competition

SINCE KUNG FU IS AN EVER POPULAR topic for computer games, we've decided to give you the chance to win your own copy of US Gold's excellent Kung Fu master.

We've got 50 copies for the readers who pick out the differences between the two cartoons and get their names picked out of the bag first.

What are you waiting for?

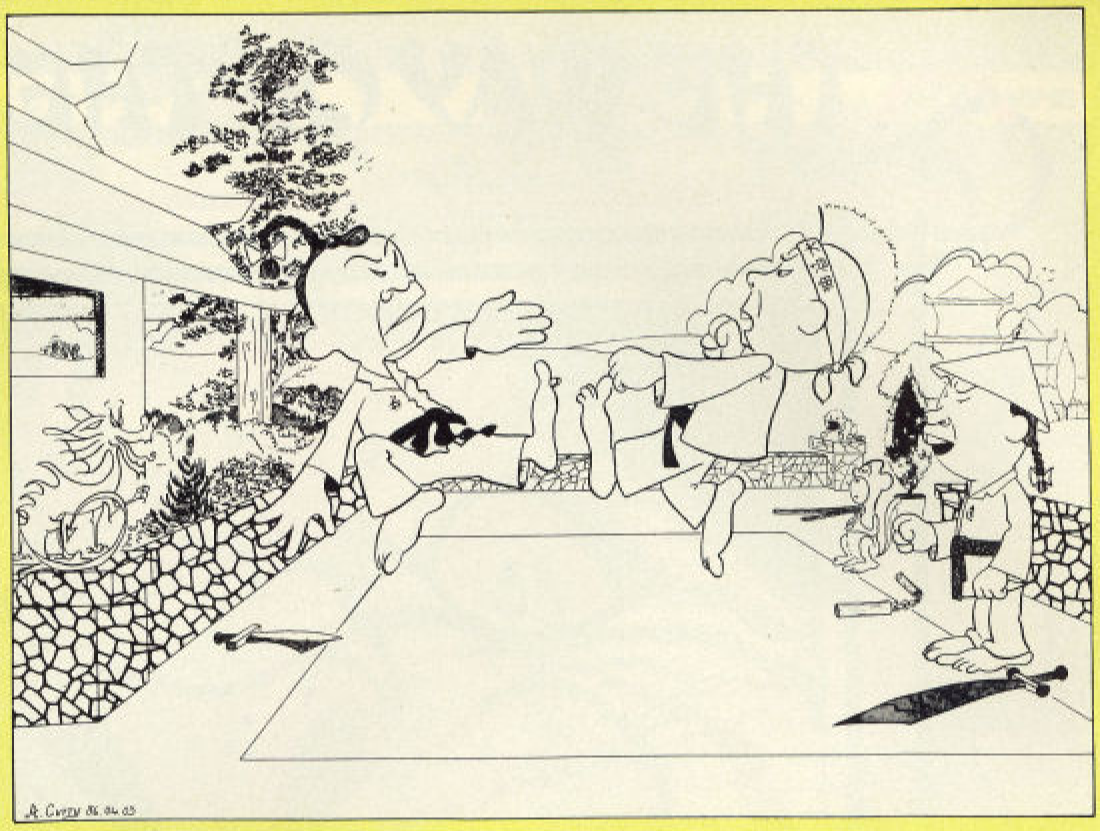
How to Enter

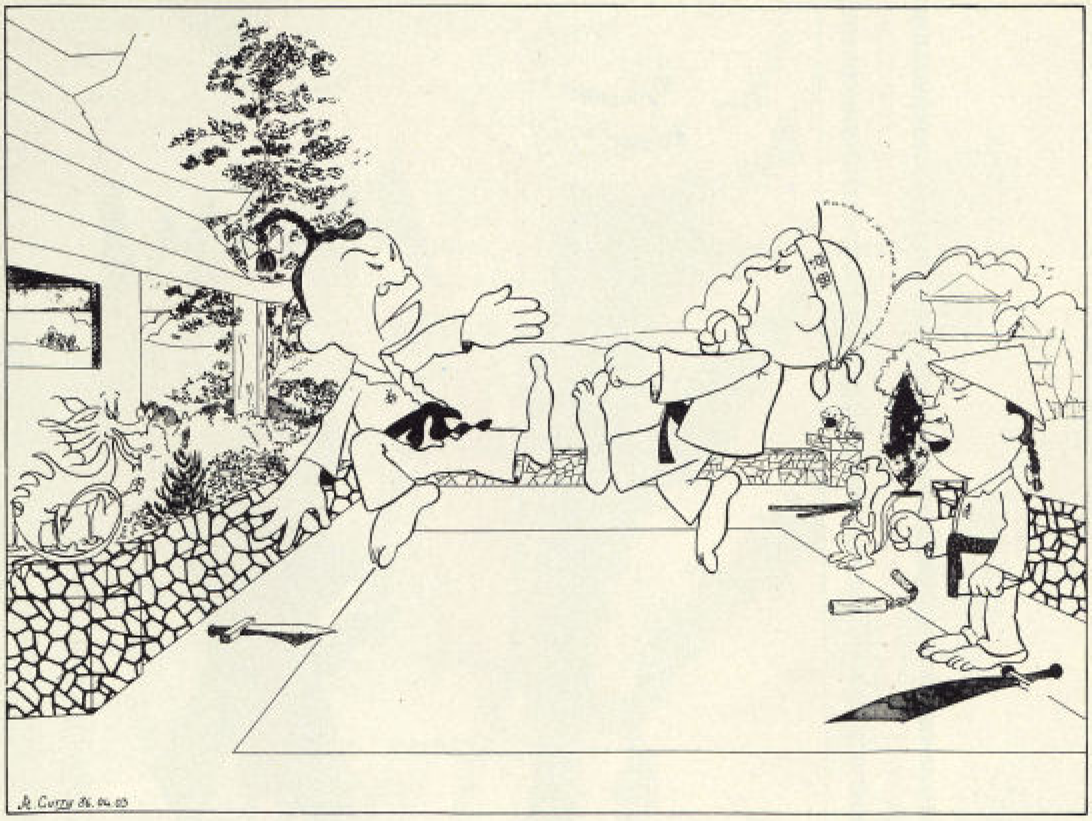
Study the two cartoons. There are several differences between them. Mark the differences clearly on the picture attached to the entry coupon. Fill in the coupon and send it off to US Gold Competition, Your Commodore, 1 Golden Square, London W1R 3AB. Write the number of differences you found on the back of your envelope.

The Rules

Entries will not be accepted from employees of Argus Specialist Publications or US Gold. This restriction applies to employee's families and agents of the company.

The How to Enter section forms part of the rules. The editor's decision is final and no correspondence will be entered into.





Entry Coupon

Name ....

Address ....

....

.... **post code**...........

Number of differences found

Send your entry to: US Gold Competition, Your Commodore, 1 Golden Square, London W1R 3AB. Closing date: Friday 27 June 1986. Write clearly and don't forget to put the number of differences on the back of your envelope.

In the first of a new series, Eric Doyle introduces you to the secret of your Commodore's memory.

CHIP CHAT

WE ALL TAKE THE CHIPS inside our computers for granted but a greater understanding of them can lead to better programs. Over the next few months I hope to clarify the function of each chip and to reveal the inner workings and hidden secrets of the Commodore range of computers: the Vic 20, C64, C-16, Plus/4 and C128.

The heart of any computer is the central processor and the most common application of the computer is to run Basic programs so this is where our Odyssey will begin.

The microprocessor in Commodore micros is a one of several derivatives of the Motorola 6500 series microchip. This processor is the number cruncher inside which are the registers that assist in all of the computer's mathematical operations. Diagram 1 shows the basic architecture of all 6500 range microprocessors.

As far as the processor is concerned the whole of the computer memory is an extension of itself from which numerical values can be loaded (read) or sent (written).

A good way of imagining memory is as a large, pigeon hole internal mailing system. Each box represents a byte of memory and can contain a value from zero to 255. Don't worry if you don't know how computers deal with numbers larger than 255, all will become clear later.

Memory comes in two varieties Read Only Memory (ROM) and Random Access Memory (RAM). As the name implies ROM can only be read from but RAM can either be read from or written to if it is necessary to change its value. For example variables defined by a Basic program must be stored in RAM for two reasons. Firstly, because it must have a value written to it to start with and, secondly, because that value may change later on in the program.

Another difference is that ROM is a permanent, non-volatile store which cannot be erased by turning the power on and off but RAM is volatile and its contents disappear when the computer is switched off.

ROM is where the Basic operating system is stored and at power up it reserves certain parts of RAM for storing the transient values generated as its routines are executed.

Processing

Load and run Listing 1 to see how the processor operates.

The program counter tells the processor where the current instruction is located in RAM/ROM memory. When commanded to execute a piece of machine code, the current value of the program counter is stored in a special reserved area of RAM memory (the processor stack area) and the internal stack pointer is adjusted to point to the next free location in the stack. The start address of the new machine code routine is then placed into the counter. This value is then loaded into the address buffer which directs the data bus to the correct location.

The data bus copies the information found in the given location and carries this back to the processor which is expecting a machine code operator. When this is evaluated in the instruction decoder it determines whether an operand or two will follow. Depending on the type of operator, any operands are evaluated and stored in the X or Y register or in the accumulator.

If the operator is a finite address the address buffer is given this value and the data bus reacts accordingly. If the value is an offset address, the finite address has the value of the X or Y register added to it and this value is passed to the address buffer.

All mathematical work is performed in the ALU which can access all of the registers which merely act as passive stores for transient values.

The instruction decoder determines whether the data bus is reading or writing its encoded information and which internal register provides or accepts the information.

When an RTS command is detected the last value to be stored on the stack is read into the program counter and the processor continues from where it was before it was asked to execute the routine.

I have said that the processor only responds to machine code routines so how does it respond to Basic?

PROGRAM: LISTING 1

```
10 CN=13:GOSUB360
20 LL=PEEK(S+1)+PEEK(S+2)*25
6
30 LC=0
40 LN=PEEK(S+3)+PEEK(S+4)*25
6:PRINT"(CLR)(WHT)LINE";LN;"
(CYN)"
50 LC=LC+1:GOSUB280:IFLC<>0T
HEN50
60 A=S+800
70 PRINT"(YEL)(DOWN)(DOWN)PL
EASE WAIT"
80 A=A+1:IFPEEK(A)=1ANDPEEK(
A+1)=143THEN100
90 GOTO80
100 PRINT"(UP)     PRESS <SP
ACE>"
110 GET A$:IF A$<>" " THEN11
0
120 S=A-4
130 LL=PEEK(S+1)+PEEK(S+2)*2
56
140 LN=PEEK(S+3)+PEEK(S+4)*2
56:PRINT"(CLR)(WHT)LINE";LN;
"":CO=0:FL=0
150 LC=LC+1:GOSUB280:IFLC<>0
THEN150
160 PRINT"00(LBLU)":LIST440-
:END
170 V$="":L=D AND 15:GOSUB20
0
180 L=(D-L)/16:GOSUB200
190 RETURN
200 IF L>9 THEN V$=CHR$(55+L
)+V$:GOTO220
210 V$=RIGHT$(STR$(L),LEN(ST
R$(L))-1)+V$
220 RETURN
230 LL=PEEK(S+LC+1)+PEEK(S+L
C+2)*256
240 FL=0:IFLL=0THEN LC=0:PRI
NT"(CYN)00 ";:GOTO270
250 PRINT:CO=0:LN=PEEK(S+LC+
3)+PEEK(S+LC+4)*256:IFLN=40T
HENLC=0:GOTO270
260 PRINT"(DOWN)(WHT)LINE";L
N;"(CYN)"
270 RETURN
280 D=PEEK(S+LC):GOSUB170
290 IF CO=2 AND FL=0 THEN PR
INT"(YEL)";
300 IF CO=4 AND FL=0 THEN PR
INT"(LBLU)";:FL=1
310 IF CO=CN THEN PRINT:CO=0
320 IF FL=1 AND D>127 AND D<
203 THEN PRINT"(GRN)"V$" (LB
LU)";:GOTO340
330 PRINT V$" ";
340 CO=CO+1:IF S+LC=LL-1 THE
N GOSUB230
350 RETURN
360 PRINT"(CLR)(DOWN)(DOWN)(
```

▶

Down to Basics

When the computer is switched on the program counter automatically loads the value it finds at $FFFC which causes the processor to run the ROM machine code routine for power reset. This organises the memory ready for Basic and ends by printing READY on the screen. The routine then loops around until a keyboard input is

After typing in or loading a program, the command RUN brings the Basic execution system into operation. To understand how this works we have to look at the way a program is stored in memory. Now enter Listing 2 to reveal the structure of a line of Basic.

The program PEEKs the program storage area and writes the actual contents of the memory locations to the screen. The first screenful of information shows the first three lines. Notice how all the lines end with a zero byte which causes the operating system to start a new line when LISTing to the screen or a printer.

The first four bytes of information have been coloured to highlight their special significance. The yellow pair of bytes gives the line number. Their actual value can be revealed in the following way:

i) Write down the first of the two bytes and then write the second byte after it (e.g. 2C 01 becomes 012C).
ii) This is the hexadecimal value of the line number. To convert it to decimal multiply the first figure by 4096, the second by 256, third by 16 and the fourth by one. Then add the new values together. (e.g. (0*4096) + (1*256)+(2*16)+(12*1)=300)

Remember that A=10, B=11 etc.

The cyan coloured figures also reveal a two byte number in the same way but this number indicates the memory location at which the next line starts. These bytes are called the line link and they help the operating system to find a particular line quickly when GOTO and GOSUB are executed or when DATA is being READ in. In other words the line links always point to the first byte of the next line link.

After these first four bytes the details of the Basic instructions follow. To make sense of

```
DOWN)(DOWN)(DOWN)(RGHT)(RGHT
)(RGHT)1) VIC 20 (3.5K)":PRI
NT"(DOWN)(RGHT)(RGHT)(RGHT)2
) VIC 20 EXPANDED
370 PRINT"(DOWN)(RGHT)(RGHT)
(RGHT)3) C128":PRINT"(DOWN)(
RGHT)(RGHT)(RGHT)4) C64":PRI
NT"(DOWN)(RGHT)(RGHT)(RGHT)5
) C16/PLUS 4
380 INPUT"(HOME)WHICH MACHIN
E (1-5)";M
390 IFM>5ORM<1THENRUN
400 IFM=1 THEN S=4096:POKE36
879,8:CN=7
410 IFM=2THENS=4608:POKE3687
9,8:CN=7
420 IFM=3THENS=7168:LISTO,1:
LIST4,1
430 IFM=4THENS=2048:POKE5328
0,0:POKE53281,0
440 REM
450 IFM=5THENS=4096:LISTO,1:
LIST4,1
460 RETURN:REM
```

```
PROGRAM: LISTING 2
```

```
10 REM ** THIS WILL NOT WORK
 ON UNEXPANDED VICS .......S
ORRY
20 REM ** FOR VIC-20 OWNERS
I HAVE LIMITED THE SCREEN SI
ZE TO 22 COLUMNS
30 REM ** OWNERS OF VIC-20S
SHOULD USE THE REMMED INSTRU
CTIONS
40 :
50 FORA=1TO33:READA$:NEXT
60 AC$="(RGHT)(RGHT)(RGHT)(R
GHT)(RGHT)(RGHT)(RGHT)(RGHT)
(RGHT)(RGHT)(RGHT)(RGHT)(RGH
T)(RGHT)(RGHT)(RGHT)(RGHT)(R
GHT)(RGHT)"
70 PRINT"(CLR)";
80 FORA=1TO15:READA$:PRINTAC
$"   "A$:NEXT
90 REM *** ON VIC-20 OMIT AC
$"   "
100 PRINT"(DOWN)(DOWN)(DOWN)
 PRESS SPACE-BAR"
110 GETA$:IFA$<>" "THEN110
120 PRINT"(HOME)";:REM *****
*****  ON VIC-20 USE PRINT"(
CLR)";
130 GOSUB720:PRINT"(HOME)":X
X=50:PA=7
140 R$="(HOME)(DOWN)(DOWN)(D
OWN)(DOWN)(DOWN)(DOWN)(DOWN)
(DOWN)(DOWN)(DOWN)(DOWN)(DOW
N)(DOWN)(DOWN)(DOWN)(DOWN)"
150 DB$(1)="  (OFF) (RVS)  "
:DB$(2)="   (OFF) (RVS) ":DB
$(3)="    (OFF) (RVS)"
160 B1$="(DOWN)(LEFT)(LEFT)(
OFF)  (RVS)":BO$="(DOWN)(LEF
T)(LEFT)  "
170 D$(1)="  "+B1$+BO$+BO$
180 D$(2)="  "+BO$+B1$+BO$
190 D$(3)="(OFF)  (RVS)"+BO$
+BO$+B1$
200 RE$="|(LEFT)(UP)↑(LEFT)(
UP)|(LEFT)(UP)":RO$=" (LEFT)
(UP) (LEFT)(UP) (LEFT)(UP)(L
EFT)| (LEFT)"
210 FORA=1TO6:RE$(A)=RE$+RE$
(A-1)
220 IFA=3THENRE$(3)=RE$(3)+"
|(LEFT)(UP)":RE$(0)=RE$(0)+"
 (LEFT)(UP)"
230 RE$(0)=RE$(0)+RO$:NEXT
240 FORA=1TO6:RE$(A)=R$+"(DO
WN)(DOWN)(DOWN)(DOWN)(RGHT)"
+AC$+RE$(A)+"(LEFT)←┐":NEXT
250 RE$(0)=R$+"(DOWN)(DOWN)(
DOWN)(DOWN)(RGHT)"+AC$+RE$(0
)
260 M$(1)="←──┘(UP)(LEFT)|(U
P)(LEFT)|(UP)(LEFT)┌─":M$(2)
="←──┐(DOWN)(LEFT)└─"
270 M$(0)="|   "+RO$+"  |(DOWN
)(DOWN)(DOWN)(DOWN)(LEFT)(LE
FT)(LEFT)   |"
280 FORA=0TO2:M$(A)=R$+"(UP)
(UP)(UP)"+LEFT$(AC$,7)+M$(A)
:NEXT
290 B=17:FORA=1TO6
300 AC$(A)=LEFT$(R$,B-3*(A-1
))+"(DOWN)"+RIGHT$(AC$,14)
310 IFA=2THENB=16
320 NEXT
330 FORA=1TO3:AC$(A)=LEFT$(A
C$(A),LEN(AC$(A))-ABS(INT(1-
4/A))):NEXT
340 MD$=LEFT$(R$,14)+"(RGHT)
(RGHT)":RO$="|(DOWN)(LEFT)":
YA$=AC$+"─┐(DOWN)(LEFT)"
350 FORA=1TO6:DP$=DP$+RO$:NE
XT:AB$=YA$+DP$+DP$:DP$=YA$+D
P$:YA$=AB$
360 AB$=LEFT$(R$,8)+AB$:DP$=
LEFT$(R$,11)+DP$+"(LEFT)─┘":
YA$="(HOME)(DOWN)"+YA$
370 ST$=LEFT$(AC$,9)+"+"
380 :
390 REM************ MAIN PROG
 ***********.
```

this note tht the green figures are tokenised keywords. Yes, a word like PRINT is converted into a single figure when the line is encoded but a variable such as SB would occupy two bytes corresponding to the CHR$ (or ASCII) values for each letter.

Another point worthy of note is that arithmetic operators have a value which is at variance with their ASCII values. When words such as PRINT or arithmetic signs are used within quotes they are not given their special status and are just evaluated as normal ASCII characters.

Slow Motion

Now back to the question of how a program RUNs.

First of all the operating system causes the processor to check if there is a program in memory. If the first two bytes of user RAM have a value then the system assumes a program is in residence. These two bytes are assumed to be a line link and are stored away for reference. The whole line up to the link location is copied into a special area of memory for analysis. This is the Basic buffer. Similarly the line number bytes are stowed away elsewhere in RAM.

After the line number the system expects to find a command of some sort. Evaluation of the token value representing the command (reserved word) is found by storing the value in the processor's accumulator and comparing this value with a list of values stored in Basic ROM. If no match is found the line is scanned in the buffer by sequentially loading each byte into the accumulator to see if there is an equals sign. The system is also comparing the accumulator's value with the token values of various punctuation marks or looking for the end-of-line zero byte. If one of these is found instead of the equals sign an error is signaled.

When such an error is indicated a SYNTAX ERROR IN LINE message is printed on the screen by loading each letter into the accumulator and moving it into screen display RAM. Next, the line number stored in RAM is dragged out, converted into a decimal value and printed after the message.

If the system has detected a

variable being created it checks the syntax of the variable name. Then an area of RAM is used to store the value along with the variable name.

If a reserved word has been found the position of the token value in the ROM list determines where the program jumps to in the Basic ROM to verify the correct syntax of any characters which follow the word until a colon or zero byte is discovered. For example, if the line was PRINT "HELLO", the PRINT token would be 99 Hex. The system would then expect a variable name, a quotation mark, a colon or a zero. In this case it finds a space, which is ignored, followed by a quotation mark. The system then takes the ASCII code from between the quotes one by one. The ASCII is converted into a screen POKE value and this places the letters which spell HELLO on the next available screen location.

When the end of a line is found the line link is retrieved from memory and the next line link is stored along with the new line number. The line is then copied into the Basic buffer and the line is scanned and interpreted as before.

This process continues until there are no more lines left or an END or STOP command is located.

Obviously with all this memory movement, evaluation and syntax checking going on the program runs a lot more slowly than a pure machine code program written specifically to create the same effect but it should be remembered that all programs are executed by the microprocessor in machine code, Basic is only a figment of the ROM's memory. When a language such as Logo or Pascal is loaded the Basic ROM is switched out in favour of the new machine code routines which interpret the syntax of the new language.

Any language on a computer is a language of convenience rather like Margaret Thatcher and President Mitterand discussing policies through the medium of an interpreter. As we all know this not only lengthens the time taken for the discussion but can also lead to misunderstandings unless great care is taken.

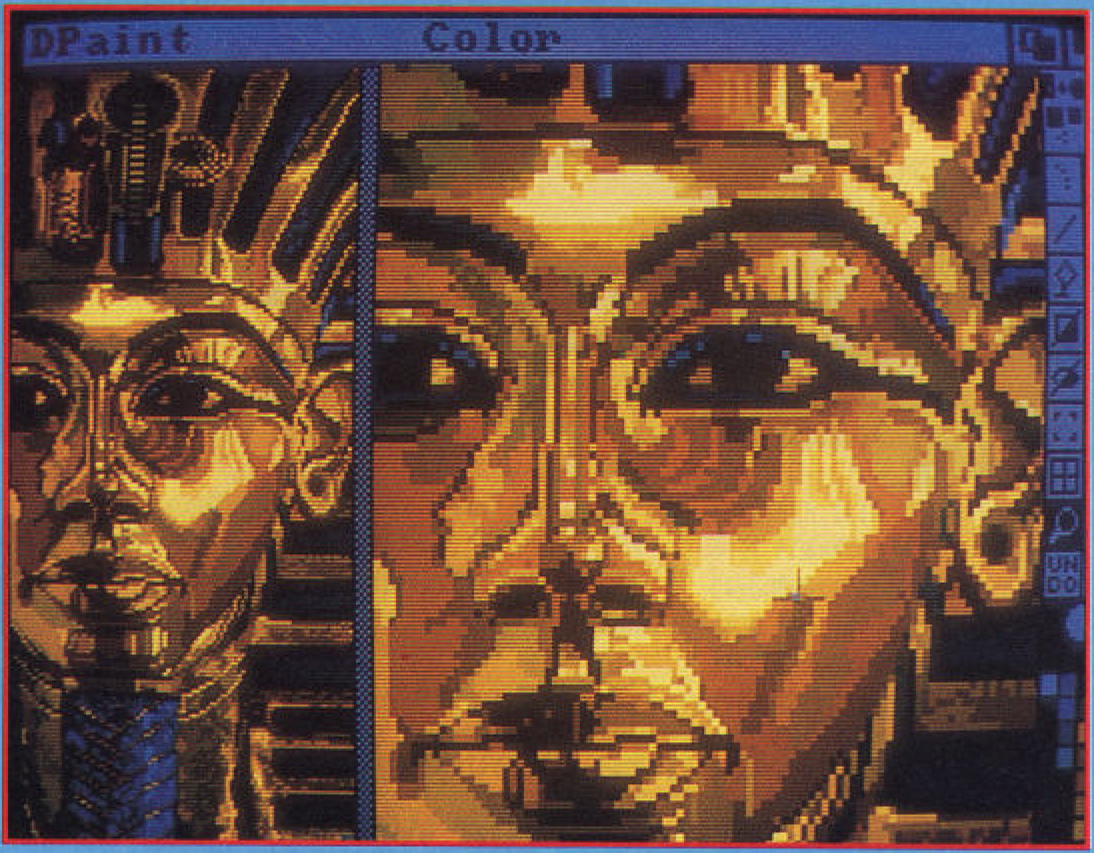
Next month we shall see how the processor helps to store numerical and string variables.

```
400 FORL=1TO4:GOSUB690:PRINTRE$(W):GOSUB1150:NEXT:REM LDY#07:LDA#20
410 FORBL=7TO0 STEP-1:YP$="(LEFT)─┘":RESTORE:BL$=RIGHT$(STR$(BL),1)
420 IFBL=6THENXX=0:PA=3
430 IFBL=5THENPA=0
440 FORL=0TO2:GOSUB690:PRINTRE$(W):GOSUB1150:NEXT:REM STA 0800,Y
450 GOSUB710:C$="(RGHT)(RGHT)0"+BL$
460 PO$=AC$(6)+"(LEFT)(LEFT)"+C$:GOSUB1190
470 READW:GOSUB1150:PRINTRE$(0)
480 PRINTM$(2):PO$=MD$+"080"+BL$:GOSUB1190
490 C$="(RVS)"+MID$("1319133439313532",BL*2+1,2):GOSUB1120
500 T$="(HOME)$080"+RIGHT$(STR$(BL),1)+"(DOWN)(LEFT)(LEFT)(LEFT)(LEFT)(LEFT)"+RIGHT$(ST$,BL+3)
510 PRINTAB$:PO$=AC$(4)+"20":GOSUB1190
520 GOSUB1000:PRINTRE$(0):GOSUB1000:READC$:GOSUB1120
530 PRINT"(HOME)(DOWN) ┌────────┐"
540 PO$=T$:GOSUB1190
550 PRINT"(HOME)(DOWN)(DOWN)";RIGHT$(AC$,BL+2);" "
560 GOSUB690:PRINTRE$(W):GOSUB1150
570 YP$="|(DOWN)(LEFT)|(DOWN)(LEFT)|(DOWN)(LEFT)(LEFT)─┘":GOSUB710
580 C$="0"+RIGHT$(STR$(BL-1),1):IFBL=0THENC$="-1"
590 READW:GOSUB1150:PRINTRE$(0)
600 FORL=0TO1:GOSUB690:PRINTRE$(W):GOSUB1160:NEXT
610 IFBL=0THENREADA$,A$:XX=50:PA=7:GOTO630
620 READC$,W:GOSUB1160
630 GOSUB960:PRINTDP$:FORD=0TOXX*10:NEXT:PRINTRE$(0):NEXTBL
640 GOSUB690:PRINTRE$(W):GOSUB1150
650 PRINT"(HOME)(DOWN)(DOWN)(DOWN)(DOWN)(RGHT)(RGHT)READY":XX=250
660 PO$="(HOME)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(RGHT)(RGHT) ":GOSUB1190
670 PRINT"(CLR)"
680 END
690 GOSUB1060:GOSUB1100:READC$:GOSUB1120:FORA=0TO1:GOSUB990:NEXT:READC$,W
700 RETURN
710 PRINTYA$+YP$:GOSUB960:RETURN
720 PRINT"$0800 (C64) ________"
730 PRINT" ┌────────┐ |  |"
740 PRINT" |SYS49152| |(RVS) Y REG (OFF)|"
750 PRINT" |        | ________"
760 PRINT" |        | |  |"
770 PRINT" |        | |(RVS) X REG (OFF)|"
780 PRINT" |        | ________"
790 PRINT" |        | |  |"
800 PRINT" |        | |(RVS) ACC  (OFF)|"
810 PRINT" └─(RVS)──────(OFF)─┘ ________"
820 PRINT"  |(RVS)SCREEN(OFF)  |(RVS) C000  (OFF)|"
830 PRINT"              |(RVS)PROGRAM(OFF)|"
840 PRINT" ______       |(RVS)COUNTER(OFF)|"
850 PRINT"|      |      ________"
860 PRINT"|      |    |  |"
870 PRINT" (RVS)  (OFF)────  |(RVS)ADDRESS(OFF)|"
880 PRINT"               ________"
890 PRINT" (RVS)  (OFF)____ |        |"
900 PRINT" (RVS)  DATA(OFF)| |(RVS) DECODER (OFF)|"
910 PRINT"   (RVS) BUS(OFF)|"
920 PRINT" (RVS)    (OFF) (RVS)    (OFF) (RVS)    (OFF) (RVS)    (OFF)"
930 PRINT" (RVS)    (OFF) (RVS)    (OFF) (RVS)    (OFF) (RVS)    (OFF)"
940 RETURN
950 REM **** DELAY
960 FORD=0TOXX:NEXT
970 RETURN
980 REM **** DATA BUS
990 FOR B=1TO3:GOTO1010
1000 FOR B=3TO1STEP-1
1010 PRINTR$;"(RGHT)(RVS)";D$(B)
1020 FORC=1TO2:PRINT"(RGHT)(RVS)"DB$(B)DB$(B)DB$(B)DB$(B):NEXT
1030 GOSUB960:NEXT
1040 RETURN
1050 REM **** PC VALUE
1060 READP$
1070 PO$=AC$(3)+"(LEFT) "+P$+" ":GOSUB1190:PRINT"(RVS)"PO$
1080 PRINTM$(1):RETURN
1090 REM **** MEMLOC
1100 GOSUB960:PRINTMD$P$:RETURN
1110 REM **** MEMCONT
1120 PO$=MD$+"(DOWN)(RGHT)"+C$:GOSUB1190
1130 PRINTM$(0):RETURN
1140 REM **** DECODER
1150 IFW=1THENPRINTAC$(W)"     "
1160 PO$=AC$(W)+C$:GOSUB1190
1170 PRINTRE$(0):RETURN
1180 REM **** FLASH
1190 FORA=0TOPA:PRINT"(RVS)"PO$"(OFF)":GOSUB960:PRINTPO$:GOSUB960:NEXT:RETURN
1200 REM **** DATA
1210 DATAC004,99,"STA    ,Y",1,C005,00," 00",2,C006,08,08,2
1220 DATA2,20
1230 DATAC007,88,"DEY      ",1,6
1240 DATAC008,10,BPL,1,C009,FA,"(RGHT)(RGHT)(RGHT)(RGHT)$FA",1,"(RGHT)(RGHT)(RGHT)$C004",1
1250 DATAC00A,60,"RTS      ",1
1260 DATACPU DEMONSTRATION,"────────────────────"
1270 DATATHIS PROGRAM WILL,SHOW HOW THE CPU,EXECUTES THE
1280 DATAFOLLOWING ROUTINE,"(DOWN)C000 LDY# $07",C002 LDA# $20
1290 DATA"C004 STA  $0800,Y",C007 DEY,C008 BPL  $C004,C00A RTS
1300 DATA"(DOWN)THIS WOULD CLEAR",PART OF THE,SCREEN ON A C64.
1310 DATAC000,A0,LDY#,1,C001,07,07,6
1320 DATAC002,A9,LDA#,1,C003,20,20,4
```

■

AMIGA

Francis Jago goes deep into the insides of the Commodore Amiga.

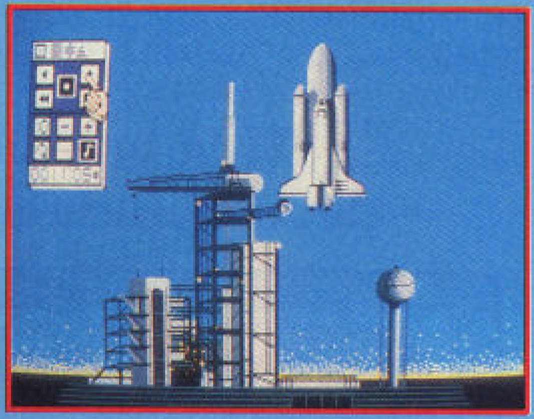


Whenever the word 'Amiga' is mentioned, someone will have something to say about Commodore's wonder-baby. At first all that was said was how wonderful this machine was, and how it would take the world by storm; gradually, however, people saw the Atari 520ST, at less than half the price — and started wondering.

I have owned an Amiga for nearly six months now, and in that time I think it is safe to say that I have formed a very personal view about a machine which, if used to its potential, can really show the true power of a personal computer.

What makes the Amiga so different from all the other personal computers is its sheer versatility. Being a true multi-tasking machine means that it can do almost as many things as you want simultaneously, enabling you to jump from one task to another with the knowledge that all the other tasks will continue.





Hardware

To successfully explain the Amiga you must really split the hardware into different sections; specification, workbench, graphics, sound, and peripherals. Although this cannot cover everything to do with the Amiga, it should give you a valuable insight into this machine.

Specification

The specification given here is of the American Amiga, when launched in Europe it will probably come with two disk drives and 512K as standard.

Basically the components of the Amiga are:

Motorola MC 68000 16/32 bit main processor.
256K bytes of internal RAM, expandable to 512K.
256K bytes of ROM containing a real-time, multi-tasking, operating system with sound, graphics and animation routines.
Built in 3½" double sided disk drive.
Expansion port for up to 3 external disk drives with either 3½" or 5¼", double sided.
Fully programmable serial port.
Fully programmable parallel port.
Two button mechanical mouse.
Two 9 pin D type controller ports.
Detached 89-key keyboard with numeric keypad, 10 function keys and cursor section.
Ports for analog or digital RGB output, as well as composite video.
Left and right stereo audio output ports.
Expansion connector that allows you to add RAM, hard disks, or other peripherals.

Workbench

At present, to start up an Amiga system, you must first insert a disk that loads the operating system into write-protected RAM. Although this does take time, it means that in the future, when new versions are released, you will not have to mess about switching chips around. Having done this you will be requested to insert what is called a Workbench disk. This is the program that makes the Amiga so easy to use, and enables a complete beginner to start harnessing the machine's power.

Once in the Workbench, most people will immediately recognise the Macintosh-esque windows and icons, however this time they are in colour! Workbench is provided for two real reasons; firstly it lets you control the computer's functions via a mouse, and secondly it lets each individual owner customise his Amiga. Using a program called Preferences, you can choose the colour of text, the colour of the background, how sensite you want the mouse to be, and also redefine the cursor which indicates the mouse position. On the more technical side, Preferences allows you to set baud rates, and redefine the bit images required for your printer.

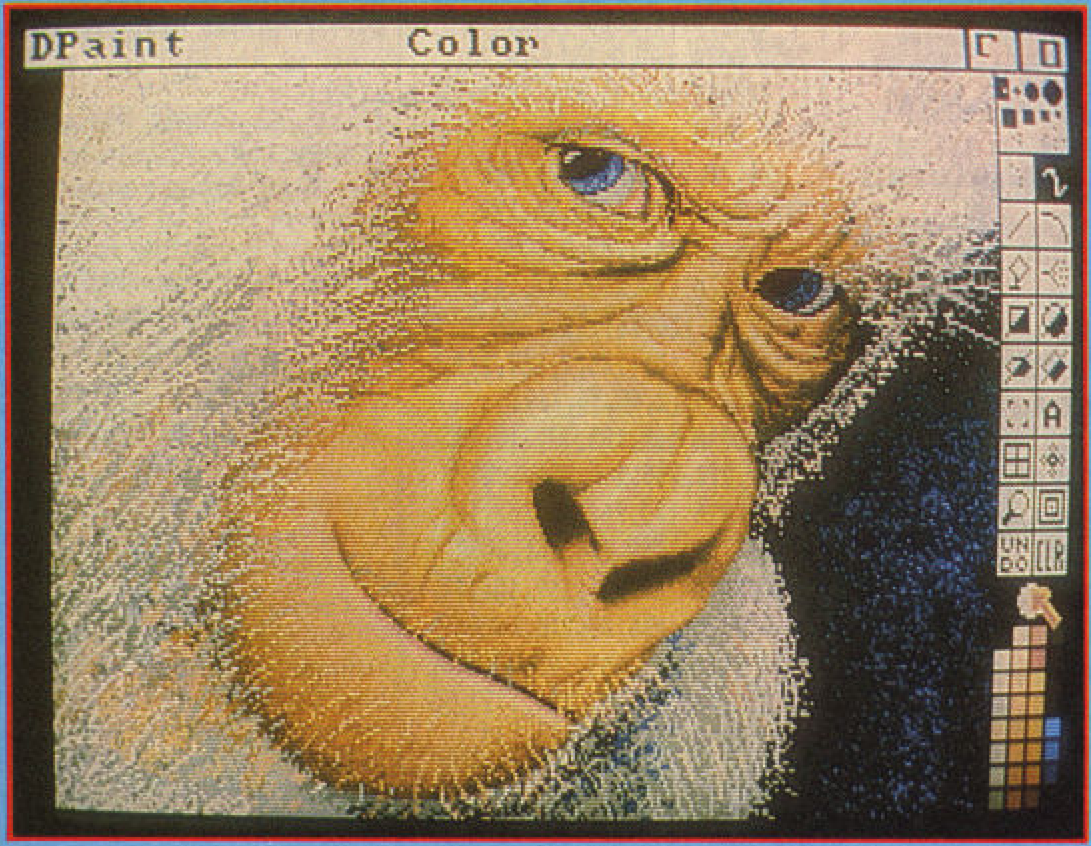
The Workbench screen, when operated, displays one large window, within which are a variety of draws. On selecting a draw with the mouse, a new window will appear, giving you another selection of Icons to choose from. Icons can best be described as small pictures which appear on the screen representing; tools, projects, disks, draws and the Trashcan. Windows let you see the contents of projects, drawers, disks, and the Trashcan. Windows can be altered both in size and position.

On the current version of the Workbench disk (v1.1) there are four draws — Demos, Utilities, System and

the detailed story







Empty. Demos are three different programs which show well how efficient a multi-tasker, the Amiga really is. Utilities gives you an on-screen calculator and notepad, à la Mac, and System produces a disk copier. The Empty draw allows you to create a personal file for the Workbench disk.

To make the most of the system, it is preferable to create your own Workbench disk to suit the type of application you will be running, I, for instance, would rather use the keyboard than the mouse, and have my Workbench set up accordingly.

Graphics

If one feature makes the Amiga stand out in a crowd more than any other it has to be the graphics. What makes this, and the sound, so outstanding are the three dedicated chips designed by Jay Miner, founder of Amiga. These chips, affectionately called AGNUS, DENISE, and PAULA, effectively allow the main CPU to do other things while they take on specific roles, such as controlling graphics and sound.

In exact terms, the Amiga has four resolutions; 320 * 200, 320 * 400, 640 * 200, and 640 * 400. However, the two modes that involve the use of 400 vertical pixels are more difficult to control as it requires a special feature called Interlacing. This allows the programmer to utilise the unused spaces to double the vertical resolution.

In each different mode you also have different amounts of available colours. In total the Amiga has a palette of 4096 colours, and in low-resolution you can put up to 32 different colours on screen at once. However, the higher resolution modes have correspondingly less available colours. One clever trick that can be used in certain situations however is called HAM (Hold And Modify). This method allows all 4096 colours to be displayed on screen simultaneously, while only sacrificing 48K. By producing this quality of colour resolution you can produce pictures of a standard as yet unsurpassed on a personal computer.

One word that will be recognisable to almost all of you is sprites. The Amiga can cope with up to eight sprites on screen at once, each of which can be as tall as is required, although only 16 pixels across. For animation purposes the Amiga also has something called a Blitter (Block Image Transferrer), and although this is by no means limited to graphics, it can be used to move large amounts of graphics data around the screen at amazing speeds, creating some outstanding effects.

It is features such as the colour palette, as well as the Blitter, that make it obvious what potential the Amiga has as a graphics machine.

Sound

To complement the Amiga's graphics, it comes with a dazzling potential for sound generation. Controlled by the Paula chip, it can produce stereo output through the left and right external sockets, and without too much difficulty, can produce sounds to rival some more expensive synthesisers.

The Amiga provides the user with four separate sound channels, each of which can be used to carry a wide range of sounds, they do not have to be monophonic. By using digital sounds and envelopes, the Amiga is quite capable of producing sounds which have been sampled, and then converted to the correct format, only to reproduce them perfectly later.

As well as producing excellent quality sound, the standard Amiga can produce quite breathtaking speech, simply by using simple commands, thus narrating software packages are no longer a thing of the past!

Peripherals

If you want a machine that will grow as a system, then the Amiga certainly has the potential. With its plethora of ports (!), it should be possible to interface almost anything to this machine, with the right software.

Printers are well provided for in the Preferences program, with most popular makes such as Epson, Diablo, Commodore, and others all having software already written to take advantage of the graphics.

Modems too are easy to rig up, with a totally programmable serial port, it should just be a question of plug in and go. I am currently running a 1200 baud modem with no troubles.

Conclusion

Although I have only managed to touch the surface of what the Amiga is really capable of, it is clear to me that, if it is marketed properly, this machine could succeed by creating a market, rather than fitting in as a run of the mill PC, that would be a real shame!

64 DATABASE

Gareth Thomas brings you your own type-in C64 Database.

The M/c Loader

MICROFILE64 IS A COMPLETE database creation and management system for the C64. It consists of a suite of three programs, the first program is mainly a machine code loader but it also carries out 'once only' operations such as setting up the SID and VIC chips etc. for the main programs, and then auto-runs the management program. The management program consists of shell-metzner sorting routines, a very powerful data search and edit system and a unique data access method for tape and disk. The third program is the format program, this allows the user to design a complete screen layout for the input of data using any of the available colours, rvs video, graphics characters and even the fullscreen editing facilities of the 64. Then by using a sprite cursor and an overlayed pop-up menu, it defines the parameters affecting each of the fields i.e. length, datatype, input position and name.

The m/c is placed above Basic in the 4K block at $C000 from $CA43 to $CAC5. So taking no Basic memory. The m/c is wedged between the stored screen definition which is split into two; the screen memory from $C000 and the colour lmemory from $CC05 using m/c routines the screen can be stored and recalled instantly for updating the file. Being stored in this block means that over 28.5K is free for data. This is allocated to 200 records of up to nine fields each one being allowed a maximum length of 255 characters (this number of records could easily be altered up to two or three times more and although the maximum field length is 255 this would probably never be used and is set at this to impose no constraints on the user. Since string storage is dynamic, the field length could be set to this and it would not immediately be allocated 255 bytes. Details on how to extend the program will be given later in the documentation).

The leader itself stores 5 m/c routines. They are:

1. Interrupt driven routine ($CAA8 - CB43): To control the Basic input routine and use the full cursor editing. This has three purposes: to stop the cursor scrolling the screen by denying it access to the top and bottom lines; de-activate the colon and comma keys while allowing their shifted, controlled and CBM functions to be registered; allows the INPUT routine to be exited by pressing function keys as well as RETURN.

2. Input routine ($CAA8 - $CB43): This routine simulates the Basic input routine but improves on it by adding: the ability to preset the maximum number of characters to be entered; allowing only certain data to be entered which can be preset and can be of four types – alphanumeric, digit, numeric or for any other numeric data being entered it is possible to specify a particular amount i.e. less than, greater than or between.

3. Cursor AT($CB46 - $CB72): This simulates the PRINT AT command found in many other Basics. It uses the kernal PLOT routine at $FFF0 and a few Basic ROM calls to allow parameter passing of the row and column positions from the SYS call.

4. Screen sdtore ($CB73 - $CBBB): This stores a copy of the screen layout defined by the format program (except for the top and bottom rows). It stores the screen memory at $CC05, the colour memory at $C000.

5. Screen recall ($CBBC - $CC04): This recalls a copy of the screen stored by the routine above.

If you wish to use the system on disk then the three programs must be saved under the filenames MF64.M/C, MF64. MAIN and MF64. FORM in order to maintain the compatibility with the routines used in the main and format programs. Also certain lines need to be altered in the machine code loader so that the main program is auto-run correctly, this is shown below.

Type in the following lines, with the m/c loader in memory:

PROGRAM: MF64.M/C

```
0 REM[SPC2]*****************
   ****
10 REM *[SPC2]MICROFILE64
   M/C[SPC2]*
20 REM *[SPC6]VER1.0[SPC7]*
30 REM * (C) 1985 G.THOMAS
   *
40 REM **********************
50 :
60 REM LOCK INTO CHARACTER
   SET 2
70 PRINT CHR$(14)CHR$(8)
80 REM BLACK SCREEN
90 POKE 53280,0:POKE 53281,0
100 :
120 PRINT"[CLEAR]"TAB(12)"
   [WHITE,RVSON,DOWN5]
   LOADING DATA
130 A=51779:LN=2000
140 READ B:IF B<256 THEN POK
   E A,B:T=T+B:A=A+1:GOTO 140
150 IF T=B THEN T=0:LN=LN+10
   :GOTO 140
160 IF B=999 THEN 190
180 PRINT TAB(9)"[DOWN3]DATA
   ERROR IN LINE"LN:STOP
190 POKE 650,128:SD=54272
   :VC=53248
200 REM SET UP SID AND VIC
   CHIPS
210 FOR L=SD TO SD+24
   :POKE L,0:NEXT
220 POKE SD+24,15
230 POKE SD+5,64:POKE SD+6,64
240 POKE SD+2,64:POKE SD+3,2
250 POKE SD,2:POKE SD+1,20
260 POKE 2040,13:POKE VC,24
270 POKE VC+21,0:POKE VC+1,58
280 POKE VC+39,1
290 REM AUTO-RUN MAIN PROGRAM
310 POKE 631,131:POKE 198,1
   :NEW
320 :
330 REM M/C DATA
2000 DATA 165,203,170,173,
   141,2,201,1,240,9,138,1443
2010 DATA 201,45,240,54,201,
   47,240,50,138,201,1417
2020 DATA 51,240,45,201,4,
   240,8,201,5,240,1235
2030 DATA 4,201,6,208,9,133,
   254,169,1,133,1118
2040 DATA 203,76,72,235,165,
   214,201,1,208,7,1382
2050 DATA 169,2,133,214,76,
   72,235,201,23,208,1333
2060 DATA 249,169,22,133,214,
   76,72,235,169,127,1466
2070 DATA 141,,220,96,120,
   169,67,141,143,2,1099
2080 DATA 169,202,141,144,2,
   88,96,120,169,72,1203
2090 DATA 141,143,2,169,235,
   141,144,2,88,96,1161
2100 DATA 169,185,32,210,255,
   32,228,255,240,251,1857
2110 DATA 170,76,206,202,165,
   251,197,252,240,241,2000
2120 DATA 230,251,169,157,32,
   210,255,138,32,210,1684
2130 DATA 255,169,185,32,210,
   255,138,96,133,254,1727
2140 DATA 201,133,240,249,
   201,134,240,245,201,135,
   1979
2150 DATA 240,241,201,13,240,
   237,201,20,208,21,1622
2160 DATA 165,251,240,197,
   198,251,198,251,169,29,
   1949
2170 DATA 32,210,255,169,20,
   32,210,255,76,182,1441
2180 DATA 202,201,32,48,176,
   201,95,16,172,165,1308
2190 DATA 253,201,1,208,16,
   138,201,32,240,170,1460
2200 DATA 201,65,48,157,201,
   91,16,153,76,182,1190
2210 DATA 202,201,2,208,33,
   138,201,43,240,150,1418
2220 DATA 201,47,240,146,201,
   42,240,142,201,45,1505
2230 DATA 240,138,138,201,46,
   240,133,201,48,48,1433
2240 DATA 14,201,58,16,10,76,
   182,202,201,3,963
2250 DATA 240,236,76,182,202,
```

```
   76,173,202,32,121,1540
2260 DATA 0,32,253,174,32,
   138,173,32,247,183,1264
2270 DATA 165,20,201,40,144,
   3,32,72,178,72,927
2280 DATA 32,253,174,32,138,
   173,32,247,183,165,1429
2290 DATA 20,201,25,176,237,
   170,104,168,24,32,1157
2300 DATA 240,255,96,169,5,
   133,251,169,204,133,1655
2310 DATA 252,169,40,133,253,
   169,4,133,254,169,1576
2320 DATA 40,133,20,169,216,
   133,21,169,,133,1034
2330 DATA 163,169,192,133,
   164,160,,177,253,145,1556
2340 DATA 251,177,20,145,163,
   200,208,245,230,252,1891
2350 DATA 230,254,230,21,230,
   164,166,254,224,7,1780
2360 DATA 208,231,177,253,
   145,251,177,20,145,163,
   1770
2370 DATA 200,192,152,208,
   243,96,169,40,133,251,1684
2380 DATA 169,4,133,252,169,
   5,133,253,169,204,1491
2390 DATA 133,254,169,,133,
   20,169,192,133,21,1224
2400 DATA 169,40,133,163,169,
   216,133,164,160,,1347
2410 DATA 177,253,145,251,
   177,20,145,163,200,208,
   1739
2420 DATA 245,230,252,230,
   254,230,21,230,164,166,
   2022
2430 DATA 252,224,7,208,231,
   177,253,145,251,177,1925
2440 DATA 20,145,163,200,192,
   152,208,243,96,1419
2450 DATA 999
```

■

```
310 PRINT"(CLR)(DOWN)(DOWN)L
OAD"CHR$(34)"MF64.MAIN"CHR$(
34)",8":POKE198,2
320 POKE631,13:POKE632,13
330 PRINT "(DOWN)(DOWN)(DOWN
)(DOWN)RUN(HOME)";:NEW
```

The Format Program

The format program can be run by selecting option eight of the main menu of microfile64 (this is discussed fully in the section dealing with the main program). The format program transfers data to the main program by saving the data as a file called "FORMAT" which can be loaded into the main program again by using option eight. The screen data is not actually saved since this could only be lost if the computer is switched off as it is stored above Basic. The transferring of data has to be carried out this way since no CHAIN command exists in CBM Basic to control variable storage when a new program is loaded in.

When the program is run, you are asked how many fields you will need from one to nine (this maximum of nine could be increased easily if necessary. The method is described later). A field is an individual data item within a record, for example, if you wished to design a layout for a club membership booking system you might need four fields; name, address, telephone and membership number. After you've typed this in, the next step is to design the screen layout.

A cursor will start flashing near the top left, you can move it anywhere on the screen using the full editing functions of the C64. Care should be taken though, since use of the insert can cause the whole screen to move down a line which can mess up your design. Apart from colon and comma keys which cause the awful "?Extra ignored" message, all other characters are acceptable including graphics, also colour and reverse can be used in the usual way. All this means that pretty nifty displays can be designed if time and care is taken.

In the design screen mode, the top line shows the user the job for each of the function keys; F1 (Exit) allows you to start designing the database again from the beginning if you made a mistake; F3 (Clear) will clear the screen to allow you to start designing the screen layout again without having to go back to the beginning; F5 (Next) skips to the next process of defining the fields and recalls a previously defined screen, this is useful if you wish to redesign the parameters for the database without having to design the screen layout again. Once you've finished designing your screen, press RETURN. The new screen will be stored instantly, wiping over any previous definition.

The next part of the program is more complex, this is where the parameters affecting each field are defined. This is in four steps. First the input position, secondly the field datatype, thirdly the input length and last of all the name of the field, this is repeated for each field. When you reach this section, the first thing to notice is the fast flashing cursor in the top left – move this cursor to where you wish each field to be input in turn, pressing RETURN to store the position. In the top right of the screen the number of the field currently being defined is

PROGRAM: MF64.MAIN

```
0 REM ********************
1 REM *[SPC3]MICROFILE64
   [SPC4]*
3 REM *[SPC2]MAIN[SPC2]
   PROGRAM[SPC3]*
4 REM * (C)1986 G.THOMAS *
5 REM ********************
6 :
15 :
17 PRINT"[CLEAR,DOWN4,WHITE]
   "TAB(9)"***[SPC,SM,SI,SC,
   SR,SO,SF,SI,SL,SE]64 ***"
   :PRINT TAB(5)"[DOWN2,
   GREEN]VER4.2(C)1986[SPC,
   SG]ARETH[SPC,ST]HOMAS
20 PRINT TAB(8)"[DOWN3,RVSON,
   YELLOW,SPC,SA,SN,SY] [SK,
   SE,SY] [ST,SO] [SC,SO,SN,
   ST,SI,SN,SU,SE,SPC]"
   :AT=52038:WAIT 198,1:RC=1
25 :
26 REM MAIN LOOP
30 GOSUB 9500:IF W=8 THEN CLR
   :W=8
35 ON W+1 GOSUB 6000,100,30,
   1800,2500,3100,3500,8000,
   4490,5825
40 IF W=9 AND RF=0 THEN CLR
   :RC=1
50 GOTO 30
55 :
56 REM DATA I/O S/R
100 PRINT TAB(12)"[SD,SA,ST,
   SA] [SI]/[SO] [SM,SE,SN,
   SU]":PRINT TAB(12)"[CT13]"
   :PRINT TAB(9)"[DOWN]1 -
   [SPC,SS]AVE TO DISK.
120 PRINT TAB(9)"[DOWN]2 -
   [SPC,SL]OAD FROM DISK."
   :PRINT TAB(9)"[DOWN]3 -
   [SPC,SS]AVE TO TAPE.
140 PRINT TAB(9)"[DOWN]4 -
   [SPC,SL]OAD FROM TAPE."
   :PRINT TAB(9)"[DOWN]5 -
   [SPC,SE]XIT.[DOWN3]
150 GOSUB 5800:IF G$<"1"OR G
   $>"5"THEN 150
160 IF G$="5"THEN RETURN
170 PRINT"[CLEAR,DOWN,RIGHT,
   SE]NTER '[SE]' TO EXIT
   [DOWN2]":INPUT"[RIGHT,SE]
   NTER FILENAME..";F$
175 IF F$=""OR LEN(F$)>16 TH
   EN 170
176 IF F$="E"THEN PRINT"
   [CLEAR]":GOTO 100
190 PRINT"[CLEAR]"
   :ON VAL(G$)GOSUB 1000,
   1110,1200,1300,1400
   :PRINT"[CLEAR]":GOTO 100
200 :
900 REM OUTPUT FILE TO DISK
1000 OPEN 15,8,15,"IO"
   :OPEN 2,8,2,"0:"+F$+",S,W"
   :GOSUB 1350:IF EI=63 THEN
    1000
1005 IF EI THEN RETURN
1010 GOSUB 1600:CLOSE 15
   :RETURN
1020 :
1030 REM INPUT FILE FROM DISK
1110 OPEN 15,8,15,"IO"
   :OPEN 2,8,2,"0:"+F$+",S,R"
   :GOSUB 1350:IF EI THEN RE
   TURN
1120 GOSUB 1700:CLOSE 15
   :RETURN
1130 :
1140 REM OUTPUT FILE TO TAPE
1200 OPEN 2,1,1,F$:GOSUB 1600
   :RETURN
1210 :
1220 REM INPUT FILE FROM TAPE
1300 OPEN 2,1,0,F$:GOSUB 1700
   :RETURN
1310 :
1320 REM READ DISK ERROR STA
   TUS
1350 INPUT#15,EI,M$
   :IF EI<20 THEN EI=0:RETURN
1370 PRINT TAB(6)"[DOWN,
   RVSON,SPC,SD]ISK ERROR
   [SPC,DOWN2]":PRINT TAB(5)
   EI"- "M$:GOSUB 9800
1380 IF EI<>63 THEN CLOSE 2
   :CLOSE 15:WAIT 198,1
   :RETURN
1390 REM IF FILE EXITS CONFI
   RM SAVE
1400 PRINT"[DOWN2]'[SR]'TO
    REPLACE OR '[SC]
   'TO CHANGE FILENAME
```

▶

displayed. If we take the example of the club membership system, then the display might look something like this:

Club Membership System
Name -
Address -
Tel. no -
Membership no. -

Then, to define the input position for 'Name', you might move the cursor to the column next to the dash after 'Name' and then store the position by pressing RETURN.

After the input position has been set an overlayed menu will come down (if you've just typed the program in it's not a bug). This is where you select the type of data the field will hold, there are four types; alphabetic, numeric, digit and other. These options are on the menu plus another five. The extra ones are used to limit the values of data entered and all conform to the rules for 'DIGIT' data;

1 ALPHABETIC – alphabetic characters only
2 NUMERIC – 0 to 9 plus +,-,/,* and point
3 DIGIT – 0 to 9 only
4 ANYTHING – any keyboard character
5 BETWEEN – between but NOT equal to two numbers
6 < than
7 > than
8 < than or = to
9 > than or = to

To select datatype move the blue cursor with the up/down cursor key to the required type and press RETURN. If you select any datatype above four then you will be asked to input the relevant values, which must be conformed to, in the window at the bottom of the menu. If you have made a mistake in selecting the input position, pressing 'E' will return back to this part without updating the field pointer to the next field, so that you may redefine.

The third parameter to define is the length. After the datatype menu has disappeared, a cross cursor will appear next to the position you defined for the input, with a reverse video representation of the field number in the exact location. Holding down any key will move the cursor which will leave a trail of dots behind it, each representing one

```
1410 GET G$:IF G$="R"THEN 14
   50
1420 IF G$<>"C"THEN 1410
1430 GOSUB 1500:IF F$="E"THE
   N EI=1:RETURN
1440 CLOSE 2:CLOSE 15:RETURN
1441 REM SCRATCH OLD FILE
1450 PRINT#15,"S0:"+F$
   :CLOSE 2:CLOSE 15:RETURN
1451 :
1530 REM SAVE FILES
1535 REM GET DATATYPE
1600 GOSUB 9800:PRINT"[DOWN2,
   RIGHT,SF]ORMAT OR DATA?
1601 GET G$:IF G$="D"THEN 16
   30
1602 IF G$<>"F"THEN 1601
1603 REM SAVE FORMAT
1609 PRINT#2,FD:PRINT#2,"F"
1610 FOR G=0 TO 6
   :FOR H=0 TO FD
   :IF B$(G,H)=""THEN B$(G,
   H)=" "
1615 PRINT#2,B$(G,H):NEXT
   :NEXT
1620 REM SAVE SCREEN LAYOUT
1621 SC=49152:CO=52229
   :FOR I=SC TO SC+919
   :PRINT#2,CHR$(PEEK(I))CHR
   $(PEEK(CO+I-SC));
1622 NEXT:CLOSE 2:RETURN
1623 :
1624 REM SAVE DATA
1630 PRINT#2,"D":PRINT#2,RC
1631 FOR G=1 TO RC-1
   :FOR H=0 TO FD
   :PRINT#2,A$(H,G):NEXT:NEXT
   :CLOSE 2:RETURN
1632 :
1633 REM INPUT FILES
1700 IF RF THEN 1720
1701 REM SAVE FORMAT
1709 INPUT#2,FD:INPUT#2,M$
1711 IF M$<>"F"THEN PRINT"
   [DOWN3,RIGHT]* *[SD]ATA
    TYPE MISMATCH* *"
   :WAIT 198,1:CLOSE 2:RETURN
1712 GOSUB 9980:FOR G=0 TO 6
   :FOR H=0 TO FD
   :INPUT#2,B$(G,H):NEXT:NEXT
1715 REM INPUT SCREEN LAYOUT
1716 SC=49152:CO=52229
   :FOR I=SC TO SC+919
   :GET#2,A$:IF A$=""THEN A$
   =CHR$(0)
1717 POKE I,ASC(A$):GET#2,A$
   :IF A$=""THEN A$=CHR$(0)
1718 POKE CO+I-SC,ASC(A$)
   :NEXT:RF=1:CLOSE 2:RETURN
1719 REM INPUT DATA
1720 INPUT#2,M$
1721 IF M$<>"D"THEN PRINT"
   [DOWN3,SPC]* *[SD]
   ATA TYPE MISMATCH* *"
   :WAIT 198,1:RC=1:CLOSE 2
   :RETURN
1722 INPUT#2,RC:FOR G=1 TO R
   C-1:FOR H=0 TO FD
   :INPUT#2,A$(H,G):NEXT:NEXT
   :CLOSE 2:RETURN
1723 :
1724 REM UPDATE S/R
1800 PRINT"[CLEAR]":EF=0
   :IF RC=200 THEN SYS AT,6,
   12:PRINT"[SN]O MORE ROOM
    FOR DATA":WAIT 198,1
   :RETURN
1810 PRINT"[HOME,C7,RVSON,
   SPC,WHITE]F1-[SE]XIT[C7,
   SPC2,WHITE]F3-[SR]EDO[C7,
   SPC2,WHITE]F5-[SM]EMORY
   [C7]:[SR]ECORD"STR$(RC)"
   [SPC3]":SYS 52156
1820 FOR I=0 TO FD:GOSUB 2000
   :IF P=133 THEN RETURN
1825 IF A$(I,RC)=""THEN A$(I,
   RC)=" ":GOTO 1850
1826 REM REMOVE EXTRA SPACES
1830 CO=0:FOR T=0 TO-1 STEP 0
   :CO=CO+1:T=(RIGHT$(A$(I,
   RC),CO)<>" "):NEXT:CO=CO-1
1840 A$(I,RC)=LEFT$(A$(I,RC),
   LEN(A$(I,RC))-CO)
1850 NEXT:RC=RC+1:GOTO 1810
1860 :
1870 REM MAIN INPUT S/R
1880 REM SET UP ZERO-PAGE
    LOCATIONS FOR M/C INPUT
2000 Y=VAL(B$(0,I))
   :X=VAL(B$(1,I))
   :SYS AT,X-3,Y-6:POKE 251,0
   :POKE 252,VAL(B$(3,I))
2010 POKE 253,VAL(LEFT$(B$(4,
   I),1))
2011 REM PRINT CURSOR AND
    SET INPUT POSITION
2020 PRINT"[CO,UP]"
   :SYS AT,X-2,Y-6:I$=""
   :FOR T=0 TO 1 STEP 0
   :SYS 51885:P=PEEK(254)
   :IF EF THEN 2035
2021 IF P=135 THEN GOSUB 9800
   :GOSUB 2400:NEXT
2025 IF P=134 AND I THEN I$=
   I$+" ":GOSUB 2420:T=1:NEXT
   :GOTO 2000
2030 IF P=133 THEN T=1:NEXT
   :RETURN
2031 REM DELETE CHARS
2035 IF P=20 THEN I$=LEFT$(I
   $,PEEK(251)):NEXT
```

character. This gives a visual representation of the field length, but on the bottom line is a count of the number of characters. The cursor may be moved until either a length of 255 is reached, or another character is encountered in front of the cursor. RETURN sets the length and DEL allows you to move back the cursor.

The final part of this section of the program is typing in the field name. You will be prompted to do this on the bottom line after defining the length. All the processes discussed then have to be repeated for each field. As in the previous section for defining the screen layout, in this section the function keys have certain uses. They are; F1 (Exit) returns back to the beginning of the program; F3 (Clear) homes the cursor and resets the field count to one; F5 (Back) allows you to go back and redefine the previous field if you made a mistake.

When you have finished defining the database the program will ask if you wish to save the data to tape or disk. If you choose tape remember to take note of the tape count. You must also wind the tape back to the position of the main program so that it can be auto-run (of course on disk this is automatic).

Management or Main Program

When the main program has auto-run, press any key to exit the intro screen so that you are then presented with the main menu. In the top right of the screen is printed the number of empty records remaining i.e. how many more records can be input. Laid out below this are the 10 main functions of the management program which can be accessed by pressing zero to nine. When the program is first run only options one and eight can be selected since no data is present and also there is no 'format' where it can be held. The formatting can be done using option eight, which will be described later, or option one can be used to lead in a format or any data.

To describe how to use the program I'll go through each option in turn:

1 — Press this yields yet another menu with five more options all of which are really

self-explanatory. When any of these are selected you will be asked to supply a filename. This must be no longer than 16 characters. Typing 'E' will exit to the main menu. Before describing other parts of DATA I/O, I must first describe the unusual method in which the program handles data files. Files can be loaded/saved in one of two forms either program data, i.e. the actual records of data or the format data. The format data takes a while to load/save even with disk since it is itself split into two halves. Firstly, the format parameters are saved i.e. field length, name etc, and then the screen layout data which takes the time to load. Saving the data as two separate files may sound strange but it creates a very flexible system. For example, if you select load from tape and a format is already present (if not then program automatically assumes you wish to load a format), firstly, you will be prompted to enter the type of file to be loaded: "Format or data?". This means that if you select data you can load in different sets of data for the same format, so it is possible to create many data files from one format file so overcoming memory limitations. If you select format and load a new format then the data present will be erased and new data file must be loaded.

If you try to load a file which is data but you select format when prompted the computer will give an error message after reading the file header and return to the menu.

Note

The data I/O option cannot be used to load a transfer file from the format program. Option eight must be used for this.

The next, option two, is print file. I have left this unwritten since I have no printer and also there are many different types, but provision has been made for a sub-routine. If pressed the menu will just be listed again. My suggestion is to place the subroutine starting at line 4200 which means the third line number (30) in the ON... GOSUB at line 35 must be altered. To help you – the s/r at 4005 will list all the records found by a search. Also look at the view option. Finally, the

```
2040 IF P<>13 THEN I$=I$+CHR
   $(P):NEXT
2041 REM CHECK FOR EDIT FLAG
2045 IF EF AND G$<>"0"THEN T
   =1:NEXT:RETURN
2050 A$(I,RC)=I$:IF LEFT$(B$
   (4,I),1)<>"3"THEN PRINT C
   HR$(20):RETURN
2060 :
2070 REM CHECK IF INPUT IS
    IN RANGE
2300 VL=VAL(A$(I,RC))
   :K$=MID$(B$(4,I),2,2)
   :VA=VAL(B$(5,I))
2310 IF(K$="<"AND VL>=VA)OR(
   K$=">"AND VL<=VA)THEN GOS
   UB 9800:NEXT
2330 IF K$="B"AND(VL<VA OR V
   L>VAL(B$(6,I)))THEN GOSUB
    9800:NEXT
2340 IF(K$="<="AND VL>VA)OR(
   K$=">="AND VL<VA)THEN GOS
   UB 9800:NEXT
2360 A$(I,RC)=I$:PRINT CHR$(
   20):RETURN
2370 :
2380 REM FREE BYTES ROUTINE
2400 P$=STR$(FRE(0))+" BYTES
    FREE":LN=LEN(P$)
   :FOR P=1 TO LN
   :POKE 1990+P,ASC(MID$(P$,
   P,1))
2410 POKE 56262+P,1:NEXT
   :POKE 198,0:WAIT 198,1
   :FOR P=1 TO LN
   :POKE 1990+P,32:NEXT
   :RETURN
2415 :
2416 REM REDO S/R
2420 A$(I,RC)=I$:D=I:ER=RC
   :GOSUB 2970:PRINT L$:I=I-1
   :D=I:GOSUB 2970:PRINT L$
   :RETURN
2430 :
2440 REM EDIT S/R
2500 PRINT TAB(13)"[SE,SD,SI,
   ST] [SM,SE,SN,SU]"
   :PRINT TAB(13)"[CT9]"
   :PRINT TAB(8)"[DOWN]1 -
   [SPC,SE]DIT RECORDS.
2530 PRINT TAB(8)"[DOWN]2
    -[SPC,SS]EARCH & REPLACE
   .":PRINT TAB(8)"[DOWN]3
    -[SPC,SS]EARCH & DELETE.
2550 PRINT TAB(8)"[DOWN]4
    -[SPC,SB]LOCK DELETE."
   :PRINT TAB(8)"[DOWN]5 -
   [SPC,SE]XIT.[DOWN3]
2590 GOSUB 5800:IF G$<"1"OR
   G$>"5"THEN 2590
2600 IF G$="5"THEN RETURN
2610 PRINT"[CLEAR]"
   :ON VAL(G$)GOSUB 2620,
   3000,3093,2890
   :PRINT"[CLEAR,WHITE]"
   :IF RC>1 THEN 2500
2615 SYS AT,8,10:PRINT"
   [RVSON,SPC,SN]O RECORDS
    REMAINING ":GOSUB 9800
   :WAIT 198,1:RETURN
2620 SYS AT,10,8:INPUT"[SR]
   ECORD NO.";ER:IF ER<1 OR
   ER>RC-1 THEN 2620
2630 M$="[C7,RVSON,SPC,WHITE]
   <-[SB]ACK[C7,SPC,WHITE]>-
   [SF]ORWARD[C7,SPC,WHITE,
   SA]-[SA]LTER[C7,SPC,WHITE,
   SD]-[SD]EL[C7,SPC,WHITE,
   SE]-[SE]XIT[C7,SPC,HOME]"
   :GOSUB 3420
2631 IF G$="A"THEN 2640
2635 IF G$="E"THEN RETURN
2636 GOSUB 2830:IF RC>1 THEN
    POKE 198,1:POKE 631,60
   :GOTO 2630
2637 RETURN
2640 GOSUB 9800:PRINT"[HOME,
   RVSON,C7,SPC5,WHITE,SPC,
   SS]ELECT DATA AND PRESS
   [SPC,SR,SE,ST,SU,SR,SN,
   SPC,C7,SPC4]"
2650 SYS AT,0,24:PRINT"[C7,
   RVSON,SPC,WHITE,SU]
   SE CURSOR KEY TO MOVE UP/
   DOWN[C7,SPC,WHITE,SE]-[SE]
   XIT[C7,SPC,HOME]
2655 D=0:GOSUB 2980
2660 GET G$:IF G$="E"THEN 26
   30
2680 IF G$="[UP]"THEN GOSUB
   2985:D=D-1:IF D<>-1 THEN
   GOSUB 2980
2690 IF D<0 THEN D=FD
   :GOSUB 2980
2700 IF G$="[DOWN]
   "THEN GOSUB 2985:D=D+1
   :IF D<>FD+1 THEN GOSUB 29
   80
2710 IF D>FD THEN 2655
2720 IF G$<>CHR$(13)THEN 2660
2730 GOSUB 9800:PRINT"[HOME,
   RVSON,C7,SPC4,WHITE,SPC,
   SS]ELECT EDIT OPTION-
   [SPC11,C7,SPC4]"
2740 SYS AT,0,24:PRINT"[C7,
   RVSON,SPC2,WHITE,SPC,SO]-
   [SO]VERWRITE[SPC,C7,SPC2,
   WHITE,SPC,SD]-[SD]ELETE
   [SPC,C7,SPC,WHITE,SPC,SE]
   -[SE]XIT[SPC,C7,SPC2,HOME]
2750 GET G$:IF G$="E"THEN 60
```

▶

array B$ holds the format data and B$ (field number two) contains the name of each field and records are held by A$ in the form A$ (field number, record number).

The third option – Update – allows the updating of the file i.e. typing in new records. When pressed, the layout you designed using the format program, should be displayed and a blue non-flashing cursor will be at the first field. Type in the necessary data and press RETURN to store and move to the next field. In the top right hand corner, a number displays the current record being typed in, as usual the function keys have been used which are also displayed on the top line they are: (Exit) will exit back to the main menu, this will not update the record counter and any data that was typed for the current record will be last; F3 (REDO) this allows the user to retype the previous field, while deleting any data that has been typed on the current field; F5 (Memory) displays on the bottom line the number of free bytes remaining, this may take a few seconds and is due to FRE(0) and not to my program. The number will remain on the bottom line until any key is pressed.

The fourth option – Edit – is probably the most complex in the program but has some very powerful function. When pressed you are presented with another menu. Again I will go through these options in order starting with Edit. After selecting this and entering the number of the record you wish to edit, once entered, the screen will display the record, and the bottom line will list the key function. The top right shows the record currently being displayed using the inequality keys (<, >). You can look at other records going backwards/forwards and when you have definitely found the record you wish to edit, pressing 'D' will delete the whole record, 'E' will exit back to the menu or 'A' will allow you to alter it. Selecting 'A' changes the prompts on the bottom and top lines and also turns the data in the first field to inverse video. Using the cursor up/down key, you can select the field to alter when you have it, pressing RETURN will change the prompts again for the third and final time asking you to select the 'Edit option', D will delete all the data in the

field or 'O' will allow you to overwrite new data into it. If you made a mistake selecting the field, E will let you exit back to choose another field.

Next on the EDIT menu is SEARCH and REPLACE. This is a very powerful feature using fully all the SEARCH routines in the program. After selecting the field you wish to search, type in the search data. The string must be prefixed with character to indicate which type of search is to be done. The six different types are listed at the top of the screen (for information on these see the documentation on for the search option).

If scan string is used it can be made to insert data in a number of records e.g. to insert a middle name into MARK ANTHONY you would type;

Search data ?@ANTHONY
Replace data?ANDREW ANTHONY

The name ANDREW would then be inserted at every occurrence of this name, but be careful since the routine searches for the first occurrence of the string within a field, i.e. if you wanted to insert my middle name DANIEL into GARETH THOMAS you could not use the shortened version e.g.

Search data?@TH
Replace data?DANIEL TH

since the result would be GAREDANIEL TH THOMAS because TH occurs twice in my name, instead you would have to type the whole e.g.

Search data?@THOMAS
Replace data?DANIEL THOMAS

Search and Delete is similar in operation and replace. If you select any of the search options apart from '@' scan string the whole field will be deleted. Using scan string can enable you to delete from inside a string e.g. to delete my middle name;

Search data? @ DANIEL

Note the space before DANIEL, since otherwise the result would be GARETHTHOMAS because the new string is reassembled around the position of the old one and their are two spaces around the old one.

```
 SUB 4400:GOTO 2640
2760 REM DELETE FIELD
2770 IF G$="D"THEN A$(D,
  ER)=" ":GOSUB 4400
  :GOTO 2640
2780 IF G$<>"O"THEN 2750
2781 REM OVERWRITE FIELD-CAL
  L INPUT S/R WITH EDIT FLA
  G SET & SAVE MAIN VARIABL
  ES
2790 GOSUB 2970:EC=RC:RC=ER
  :I=D:GOSUB 2000
2800 IF I$=""THEN A$(D,ER)="
  "
2810 RC=EC:GOSUB 4400
  :GOTO 2640
2830 SYS AT,0,24:PRINT"[C7,
  RVSON,SPC6,WHITE,SPC3,SA]
  RE YOU SURE?[SPC4,C7,
  SPC12,HOME]
2840 GET G$:IF G$="N"THEN RE
  TURN
2860 IF G$<>"Y"THEN 2840
2870 SYS AT,0,24:PRINT"[C7,
  RVSON,SPC8,WHITE,SPC5,SD]
  ELETING DATA..[SPC4,C7,
  SPC4,HOME]
2871 REM MOVE RECORDS ABOVE
  DELETED ONE DOWN
2880 FOR C=ER TO RC
  :FOR F=0 TO FD
  :A$(F,C)=A$(F,C+1):NEXT
  :NEXT:RC=RC-1:RETURN
2881 :
2882 REM BLOCK DELETE S/R
2890 SYS AT,5,6:INPUT"[SD]
  ELETE FROM.."; DF
  :IF DF<1 OR DF>RC-2 THEN
  2890
2895 ER=DF:M$="[RVSON,C7,
  SPC9,WHITE,SPC,SA]RE YOU
  SURE(Y/N)?[SPC,C7,SPC10,
  HOME]":GOSUB 9800
  :GOSUB 4390
2896 GET G$:IF G$="N"THEN PR
  INT"[CLEAR]":GOTO 2890
2897 IF G$<>"Y"THEN 2896
2900 PRINT"[CLEAR]"
  :SYS AT,5,10:INPUT"[SD]
  ELETE TO....";DT
  :IF DT<=DF OR DT>RC-1 THE
  N 2900
2902 ER=DT:GOSUB 9800
  :GOSUB 4390
2903 GET G$:IF G$="N"THEN PR
  INT"[CLEAR]":GOTO 2900
2904 IF G$<>"Y"THEN 2903
2905 SYS AT,0,24:PRINT"
  [RVSON,C7,SPC,WHITE,SPC,
  SD]O YOU WISH THE DELETE
  TO GO AHEAD?[SPC,C7,SPC,
  HOME]":GOSUB 9800
2906 GET G$:IF G$="N"THEN RE
  TURN
2907 IF G$<>"Y"THEN 2906
2910 PRINT"[CLEAR]"
  :SYS AT,9,14:PRINT"$ $
  [SPC,SD,SE,SL,SE,ST,SI,SN,
  SG,SPC]$ $
2911 REM MOVE RECORDS DOWN
  TO DELETE BLOCK
2920 FOR D=DT+1 TO RC-1
  :FOR F=0 TO FD
  :A$(F,DF-1+(D-DT))=A$(F,D)
  :NEXT:NEXT
2930 RC=RC-(DT-DF+1):RETURN
2940 :
2950 REM DELETE DATA ON SCRE
  EN FOR OVERWRITE
2970 L$="":FOR L=1 TO LEN(A$
  (D,ER)):L$=L$+" ":NEXT
  :GOSUB 2995:PRINT L$
  :RETURN
2975 :
2976 REM S/R TO HIGHLIGHT
  CURRENT FIELD TO EDIT
2980 GOSUB 2995:PRINT"[RVSON]
  "A$(D,ER):RETURN
2985 GOSUB 2995:PRINT A$(D,
  ER):RETURN
2990 SYS AT,0,24:PRINT"[C7,
  RVSON,SPC,WHITE]F1-[SD]
  ELETE[SPC,C7,SPC,WHITE,
  SPC]F3-[SO]VERWRITE[SPC,
  C7,SPC,WHITE,SPC]F5-[SE]
  XIT[SPC,C7,SPC2,HOME]
2991 REM CURSOR SET S/R
2995 SYS AT,VAL(B$(1,D))-3,
  VAL(B$(0,D))-6:RETURN
2996 :
2997 REM SEARCH & REPLACE
  S/R
3000 EF=1:IS=0:GOSUB 4350
  :INPUT"[DOWN,RIGHT2,SF]
  IELD NO.(-1 TO EXIT)";SI
  :IF SI=-1 THEN RETURN
3010 IF SI<1 OR SI>FD+1 THEN
  3000
3020 S$="":SI=SI-1
  :SYS AT,2,16:INPUT"[SS]
  EARCH DATA";S$
  :IF S$=""THEN 3000
3025 SYS AT,2,18:PRINT"[SR]
  EPLACE DATA?":X=18:Y=24
  :SYS AT,15,18:POKE 251,0
3026 REM CALL INPUT ROUTINE
  TO MAKE SURE DATA CONFOR
  MS TO FIELD TYPE
3030 POKE 252,VAL(B$(3,SI))
  :POKE 253,VAL(LEFT$(B$(4,
  SI),1)):GOSUB 2020
3031 REM CALL SEARCH S/R
3040 IT$=LEFT$(S$,1)
  :LN=LEN(S$)-1:S$=RIGHT$(S
  $,LN):GOSUB 3599
3050 IF CP$=""THEN 3000
3060 PRINT TAB(10)"[DOWN2,SR]
  EPLACING..[SPC,DOWN3]"
  :FOR D=1 TO LEN(CP$)
  :P=ASC(MID$(CP$,D,1))
3070 IF IT$<>"@"THEN A$(SI,
  P)=I$:GOTO 3092
3075 REM CHECK LENGTH OF STR
  ING DOES NOT EXCEED FIELD
  LENGTH
3080 IF LEN(A$(SI,
  P))-LN+LEN(I$)<=VAL(B$(3,
  SI))THEN 3091
3090 PRINT TAB(3)"[UP]$ $
  [SPC,SS]TRING TOO LONG
  IN"P" $ $":GOSUB 9800
  :GOTO 3092
3091 A$(SI,P)=LEFT$(A$(SI,P)
  PS-1)+I$+RIGHT$(A$(SI,P),
  LEN(A$(SI,P))-(PS-1+LN))
3092 NEXT:GOSUB 9800:RETURN
3093 EF=1:IS=0:GOSUB 4350
  :INPUT"[DOWN2,RIGHT2,SF]
  IELD NO.(-1 TO EXIT)";SI
  :IF SI=-1 THEN RETURN
3094 IF SI<1 OR SI>FD+1 THEN
  3093
3095 SI=SI-1:INPUT"[DOWN2,
  RIGHT2,SS]EARCH DATA";S$
  :IF S$=""THEN 3093
3096 IT$=LEFT$(S$,1)
  :LN=LEN(S$)-1:S$=RIGHT$(S
  $,LN):GOSUB 3599
  :IF CP$=""THEN 3093
3097 PRINT TAB(9)"[DOWN2,
  RVSON,SPC,SD]ELETING..
  ":FOR D=1 TO LEN(CP$)
  :P=ASC(MID$(CP$,D,1))
3098 IF IT$<>"@"THEN A$(SI,
  P)=" ":NEXT:RETURN
3099 A$(SI,P)=LEFT$(A$(SI,P),
  PS-1)+MID$(A$(SI,P),PS+LN,
  LEN(A$(SI,P))):NEXT:RETURN
3100 PRINT"[CLEAR,WHITE,
  DOWN3,RIGHT2,SP]RESS;"
  :PRINT TAB(5)"[DOWN2]1
  -[SPC,ST]O VIEW A RECORD.
3101 PRINT TAB(5)"[DOWN2]2
  -[SPC,ST]O VIEW ALL RECO
  RDS.":PRINT TAB(5)"[DOWN2]
  3 -[SPC,SE]XIT.
3102 GET G$:IF G$="2"THEN 31
  20
3103 IF G$="3"THEN RETURN
3104 IF G$<>"1"THEN 3102
3105 PRINT"[CLEAR]
```

```
3106 SYS AT,4,8:INPUT"[SR]
   ECORD NO.";ER:IF ER<1 OR
   ER>RC-1 THEN 3106
3108 M$="[RVSON,C7,SPC7,
   WHITE,SPC,SP]RESS A KEY
    TO EXIT[SPC,C7,SPC11,
   HOME]":GOSUB 4390
3109 WAIT 198,1:GOTO 3100
3120 PRINT"[CLEAR,DOWN4,
   RIGHT4,SA,SN,SY,SPC,SK,SE,
   SY,SPC]TO SCROLL '[SE]'TO
    EXIT[DOWN2]
3150 SYS AT,6,10:INPUT"[SR]
   ECORD NO.(-1 TO EXIT)";SP
   :IF SP=-1 THEN RETURN
3160 IF SP<1 OR SP>RC-1 THEN
    3150
3165 REM DISPLAY FILES BY
    SCROLLING
3170 PRINT"[CLEAR]
3220 FOR P=SP TO RC-1
   :PRINT"[RVSON,SPC,SR]
   ECORD "P:FOR SR=0 TO FD
   :PRINT B$(2,SR)"
   : "A$(SR,P)
3230 GET G$:IF G$="E"THEN 31
   00
3240 NEXT:PRINT"- - - - -
    - - - - - - - - - - -
    - - - -[SPC,UP2]
3260 WAIT 198,1:NEXT
   :WAIT 197,64:POKE 198,0
   :WAIT 198,1:GOTO 3100
3400 SYS AT,7,3:INPUT"[SR]
   ECORD NO.";ER:IF ER<1 OR
   ER>RC-1 THEN 3400
3415 M$="[C7,RVSON,SPC4,
   WHITE,SPC]<-[SB]ACK[SPC,
   C7,SPC2,WHITE,SPC]>-[SF]
   ORWARD[SPC,C7,SPC2,WHITE,
   SPC,SE]-[SE]XIT[SPC,C7,
   SPC4,HOME]
3416 REM S/R TO DISPLAY RECO
   RDS WITHOUT SCROLLING USI
   NG SCREEN LAYOUT
3420 FOR T=0 TO 1 STEP 0
   :PRINT"[HOME,C7,RVSON,
   SPC2,SR]ECORD "STR$(ER)"
   [SPC29]"
3430 GOSUB 4400
3450 GET G$:IF G$="E"OR G$="
   D"OR G$="A"THEN T=1:NEXT
   :RETURN
3470 IF G$<>","THEN 3480
3475 ER=ER-1:IF ER=0 THEN ER
   =RC-1
3478 NEXT
3480 IF G$<>"."THEN 3450
3490 ER=ER+1:IF ER=RC THEN E
   R=1
3495 NEXT
3496 :
3497 REM SEARCH S/R
3500 EF=0:PRINT"[CLEAR]"
   :GOSUB 4350:PRINT"[DOWN,
   RIGHT3,GREEN,SE]NTER IN
    FORM,[RVSON,SF,RVSOFF]
   IELD[RVSON,SS,RVSOFF]
   EARCHTYPE[RVSON,SD,RVSOFF]
   ATA
3505 PRINT"[DOWN,RIGHT,
   YELLOW]'[SE]' TO EXIT '
   [SC]' TO CLEAR '[SV]' TO
    VIEW[DOWN2]
3510 S$="":POKE 198,0
   :INPUT S$:IF S$="E"THEN I
   S=0:RETURN
3520 IF S$="C"THEN IS=0
   :CP$="":GOTO 3500
3525 IF S$="V"AND CP$<>""THE
   N PRINT"[CLEAR]"
   :GOSUB 4005
3526 IF S$=""OR LEN(S$)<3 TH
   EN 3500
3530 SI$=LEFT$(S$,1)
   :IF SI$<"1"OR SI$>RIGHT$(
   STR$(FD+1),1)THEN 3500
3535 SI=VAL(SI$):SI=SI-1
   :GOSUB 3598:GOTO 3500
3536 :
3537 REM MAIN S/R
3598 IT$=MID$(S$,2,1)
   :LN=LEN(S$)-2:S$=RIGHT$(S
   $,LN)
3599 IF IT$<>"!"AND IT$<>"$"
   AND IT$<>"@"AND IT$<>"<"A
   ND IT$<>">"AND IT$<>"
   [UP-ARROW]"THEN RETURN
3600 IF EF THEN PRINT"[CLEAR]
   ":SYS AT,0,8
3601 PRINT TAB(5)"[DOWN,SS]
   EARCHING..
3602 IF IS=0 THEN IS=1
   :GOSUB 3800:GOTO 3621
3610 GOSUB 3810
3621 IF CP$<>""THEN 3630
3625 GOSUB 9800:PRINT TAB(5)
   "[UP]$ $[RVSON,SPC,SN]O
    DATA FOUND[SPC,RVSOFF]$
    $":GOSUB 3900
3626 IF IS THEN CP$=CN$
3627 RETURN
3630 IM=LEN(CP$):PRINT TAB(5
   )"[UP]"STR$(IM)" RECORDS
    FOUND":GOSUB 9800
3640 GOSUB 3900:RETURN
3700 :
3710 REM SEARCH SUBROUTINES
3720 REM DIRECT COMPARISON
3781 IF IT$="!"AND S$=A$(SI,
   G)THEN CP$=CP$+CHR$(G)
3782 IF IT$="[UP-ARROW]
   "AND S$<>A$(SI,
   G)THEN CP$=CP$+CHR$(G)
3783 IF IT$="<"AND S$>A$(SI,
   G)THEN CP$=CP$+CHR$(G)
3784 IF IT$=">"AND S$<A$(SI,
   G)THEN CP$=CP$+CHR$(G)
3785 NEXT:RETURN
3786 :
3787 REM WILDCARD
3790 IF LEFT$(A$(SI,G),
   LN)=S$THEN CP$=CP$+CHR$(G)
3795 NEXT:RETURN
3796 REM SEARCH STRING
3797 FOR S=1 TO LEN(A$(SI,
   G))-LN+1
3798 IF S$=MID$(A$(SI,G),S,
   LN)THEN CP$=CP$+CHR$(G)
   :G=G+1:PS=S
3799 NEXT:NEXT:RETURN
3800 CP$="":FOR G=1 TO RC-1
   :GOTO 3820
3810 CN$=CP$:CP$=""
   :FOR T=1 TO LEN(CN$)
   :G=ASC(MID$(CN$,T,1))
3820 IF IT$="$"THEN 3790
3830 IF IT$="@"THEN 3797
3840 IF IT$="<"OR IT$=">"OR
   IT$="[UP-ARROW]
   "OR IT$="!"THEN 3781
3900 FOR W=1 TO 1000:NEXT
   :RETURN
3905 :
3910 REM DISPLAY RECORDS FOU
   ND
4000 PRINT TAB(6)"[DOWN2,
   RVSON,SP,SR,SE,SS2] [SA,
   SN,SY] [SK,SE,SY] [ST,SO]
    [SS,ST,SA,SR,ST,RVSOFF]"
   :WAIT 198,1:PRINT"[CLEAR]
4005 FOR D=1 TO LEN(CP$)
   :P=ASC(MID$(CP$,D,1))
4010 PRINT"[RVSON,SPC,SR]
   ECORD "P:FOR SR=0 TO FD
   :PRINT B$(2,SR)"
   : "A$(SR,P):GET G$
   :IF G$="E"THEN RETURN
4040 NEXT:PRINT"- - - - -
    - - - - - - - - - - -
    - - - -[SPC,UP2]"
   :WAIT 198,1:NEXT
4050 WAIT 197,64:WAIT 198,1
   :RETURN
4060 :
4070 REM SEARCH INTRO S/R
4350 PRINT"[CLEAR,RVSON]$
   [RVSOFF]WILDCARD[SPC,
   RVSON]![RVSOFF]COMPARE
    DIRECTLY[SPC,RVSON]@
   [RVSOFF]SCAN STRING
```

For both edit related searches it should be noted that the first occurrence of the string is used and also that the whole of lthe file will be searched for the necessary data even if one alteration is made.

The final EDIT option is Block Delete. This allows you to delete large chunks of records quickly, just enter the numbers of the first and last records to be deleted and the program will do the rest.

Now back to the main menu. Next is the view option file. This is very simple, you can either view one record (this uses the format layout) or you can select a record to start from and scroll the whole file. If you wish to view all the records, holding down a key will cause them to scroll down the screen. If you release the key the records will pause. Pressing 'E' will exit back to the start.

Sixth on the main menu is the Search option. To make it as fast and easy as possible information is entered about the field and search type in one string e.g. to search field three comparing directly and searching for LONDON you would enter;

3!LONDON

The computer will then tell you how many records it finds but it will not display them. Instead the screen is set to its original form and you are prompted for more data. The next search done will not search all the records but it will search only these found by the previous search, this means that you can continually search a decreasing number of records using different search formats and so narrowing down the records to the few you are looking for. This is a very powerful feature if used properly. Suppose a file was created of names and addresses of people for a tennis club, you might need to search for all the people living in BATH who play singles and have a name beginning with A. Let's suppose that field one is Name, two is Address and three is type of player. You would enter 2@BATH (need to scan all of the address for the string BATH)

1*
3!SINGLES

and then enter 'v' to view.

```
4360 PRINT"[RIGHT,RVSON]<
    [RVSOFF]LESS THAN[SPC,
    RVSON]>[RVSOFF]GREATER
    THAN[SPC,RVSON,UP-ARROW,
    RVSOFF]NOT EQUAL TO[DOWN]"
    :GOSUB 5000:RETURN
4370 REM RECORD IN SCREEN
    FORMAT
4385 REM 6 AND 25 SPACES
4390 PRINT"[CLEAR,RVSON,C7,
    SPC6,SR]ECORD-"STR$(ER)"
    [SPC25]"
4400 SYS 52156:FOR D=0 TO FD
    :SYS AT,VAL(B$(1,D))-3,
    VAL(B$(0,D))-6
    :PRINT A$(D,ER):NEXT
4410 SYS AT,0,24:PRINT M$
    :RETURN
4420 :
4430 REM FORMAT S/R
4440 REM EITHER LOADS FORMAT
    DATA CREATED BY FORMAT
    PROGRAM
4450 REM OR AUTO-RUNS FORMAT
    PROGRAM
4490 AT=52038:SYS AT,6,4
    :PRINT"[SP]RESS 'E' TO
    EXIT
4495 SYS AT,4,8:PRINT"[SP]
    ROGRAM OR[SPC,SD]ATA(P/D)?
4500 GET G$:SYS AT,25,8
    :PRINT G$:IF G$="P"THEN 4
    600
4505 IF G$="E"THEN RC=1
    :RETURN
4510 IF G$<>"D"THEN 4500
4520 GOSUB 4735:IF G$<>"T"TH
    EN 4531
4521 PRINT"[DOWN2,RIGHT,SW]
    IND TO FORMAT DATA & PRES
    S[SPC,RVSON,SR,SE,ST,SU,
    SR,SN]":WAIT 197,1
    :OPEN 2,1,0,"FORMAT"
    :GOTO 4550
4531 IF G$="E"THEN RC=1
    :RETURN
4536 REM OPEN FILE TO DISK
4540 OPEN 15,8,15,"IO"
    :OPEN 2,8,2,"FORMAT,S,R"
    :GOSUB 1350:IF EI THEN RE
    TURN
4545 REM INPUT DATA
4550 INPUT#2,FD:GOSUB 9980
    :FOR G=0 TO 6:FOR H=0 TO
    FD:INPUT#2,B$(G,H):NEXT
    :NEXT:RF=1
4570 IF G$="T"THEN CLOSE 2
    :RETURN
4580 CLOSE 15:CLOSE 2:RETURN
4590 :
4595 REM AUTO-RUN MAIN PROGR
    AM
4600 GOSUB 4735:IF G$="T"THE
    N 4700
4610 GET G$:SYS AT,17,11
    :PRINT G$:IF G$="T"THEN 4
    700
4615 IF G$="E"THEN RETURN
4630 PRINT"[CLEAR,DOWN2]
    LOAD"CHR$(34)"MF64.FORM"C
    HR$(34)",8":POKE 198,2
4650 POKE 631,13:POKE 632,13
    :PRINT"[DOWN4]RUN[HOME]";
    :NEW
4700 PRINT"[DOWN2,RIGHT,SW]
    IND TO FORMAT PROGRAM &
    PRESS[SPC,RVSON,SR,SE,ST,
    SU,SR,SN]
4720 WAIT 197,1:POKE 631,131
    :POKE 198,1:NEW
4730 :
4735 PRINT TAB(4)"[DOWN2,ST]
    APE OR[SPC,SD]ISK(T/D)?
4736 GET G$:SYS AT,22,11
    :PRINT G$:IF G$<>"T"AND G
    $<>"D"AND G$<>"E"THEN 4736
4737 RETURN
4740 REM S/R TO PRINT FIELDS
5000 FOR G=0 TO FD
    :PRINT G+1"[LEFT]."B$(2,G)
    :NEXT::RETURN
5010 :
5020 REM S/R TO FLASH PROMPT
5800 PRINT TAB(11)"[UP,SP,SR,
    SE,SS2] [ST,SO] [SS,SE,SL,
    SE,SC,ST]":FOR D=1 TO 65
    :GET G$:IF G$<>""THEN RET
    URN
5810 NEXT:PRINT TAB(11)"[UP,
    RVSON,SP,SR,SE,SS2] [ST,
    SO] [SS,SE,SL,SE,SC,ST]"
    :FOR D=1 TO 65:GET G$
    :IF G$<>""THEN RETURN
5820 NEXT:GOTO 5800
5821 :
5822 REM NEW DATA S/R
5825 SYS AT,2,5:INPUT"[SD]
    ATA OR DATA & DEFINITIONS
    ([SD]/[SF])";G$
    :IF G$<>"D"AND G$<>"F"THE
    N 5825
5830 SYS AT,8,10:INPUT"[SA]
    RE YOU SURE([SY]/[SN])";Q$
    :IF Q$="N"THEN RETURN
5840 IF Q$<>"Y"THEN 5830
5850 IF G$="F"THEN RF=0
5860 RC=1:RETURN
6000 SYS AT,8,5:INPUT"[SA]RE
    YOU SURE([SY]/[SN])";Q$
    :IF Q$="N"THEN RETURN
6020 IF Q$<>"Y"THEN 6000
6030 SYS AT,3,9:INPUT"[SD]O
    YOU WISH TO SAVE DATA(Y/
    N)";Q$:IF Q$="N"THEN 6060
6050 IF Q$<>"Y"THEN 6030
6055 PRINT"[CLEAR]":GOSUB 100
6060 SYS 64738
6070 :
7830 REM SORT[SPC2]S/R
8000 PRINT"[CLEAR]"
    :GOSUB 5000
8010 INPUT"[DOWN2,SS]ORT BY
    WHICH FIELD(-1 TO EXIT)";
    SF
8020 IF SF=-1 THEN RETURN
8030 IF SF<1 OR SF>FD+1 THEN
    PRINT"[UP2]":GOTO 8010
8040 PRINT TAB(9)"[DOWN4]*
    *[SPC,RVSON,SPC,SS,SO,SR,
    ST,SI,SN,SG,SPC,RVSOFF,
    SPC]* *":N=RC-1:SF=SF-1
8050 IF B$(4,SF)="2"THEN 8200
8060 :
8070 REM ALPHABETIC SORT
8100 M=INT(N/2):FOR A=-1 TO
    0 STEP 0:FOR J=1 TO N-M
    :FOR H=J TO 0 STEP-M:L=H+M
    :K=0
8110 IF A$(SF,H)>A$(SF,
    L)THEN FOR Y=0 TO FD
    :Z$=A$(Y,H):A$(Y,H)=A$(Y,
    L):A$(Y,L)=Z$:NEXT:K=H
8120 H=K:NEXT:NEXT:M=INT(M/2)
    :A=(M>0):NEXT:GOSUB 9800
    :RETURN
8130 :
8140 REM NUMERIC SORT
8200 M=INT(N/2):FOR A=-1 TO
    0 STEP 0:FOR J=1 TO N-M
    :FOR H=J TO 0 STEP-M:L=H+M
    :K=0
8210 IF VAL(A$(SF,
    H))<VAL(A$(SF,L))THEN 8230
8220 FOR Y=0 TO FD:Z$=A$(Y,H)
    :A$(Y,H)=A$(Y,L)
    :A$(Y,L)=Z$:NEXT:K=H
8230 H=K:NEXT:NEXT:M=INT(M/2)
    :A=(M>0):NEXT:GOSUB 9800
    :RETURN
8240 :
8250 REM MAIN MENU
9500 PRINT"[CLEAR]"TAB(8)"
    [WHITE,SM,SI,SC,SR,SO,SF,
    SI,SL,SE]64[SPC,SM,SA,SI,
    SN] [SM,SE,SN,SU,SPC]
    "201-RC"FREE":PRINT TAB(8
    )"[CT21]
9520 PRINT TAB(8)"1 -[SPC,SD]
    ATA[SPC,SI]/[SO]."
    :PRINT TAB(8)"[DOWN]2 -
    [SPC,SP]RINT FILE."
    :PRINT TAB(8)"[DOWN]3 -
    [SPC,SU]PDATE FILE.
9550 PRINT TAB(8)"[DOWN]4
    -[SPC,SE]DIT FILE."
    :PRINT TAB(8)"[DOWN]5 -
    [SPC,SV]IEW FILE."
    :PRINT TAB(8)"[DOWN]6 -
    [SPC,SS]EARCH FILE.
9580 PRINT TAB(8)"[DOWN]7
    -[SPC,SS]ORT FILE."
    :PRINT TAB(8)"[DOWN]8 -
    [SPC,SF]ORMAT DATABASE."
    :PRINT TAB(8)"[DOWN]9 -
    [SPC,SN]EW DATA.
9605 PRINT TAB(8)"[DOWN]0
    -[SPC,SE]XIT PROGRAM.
    [DOWN2]
9620 GOSUB 5800:IF G$<"0"OR
    G$>"9"THEN 9620
9650 W=VAL(G$):IF RF=0 AND W
    <>1 AND W<>0 AND W<>8 THE
    N 9620
9655 IF RF AND W=8 THEN 9620
9660 IF(RF AND RC=1)AND W<>3
    AND W<>1 AND W<>9 THEN 9
    620
9670 PRINT"[CLEAR]":RETURN
9680 :
9690 REM BEEP S/R
9800 POKE 54276,65
    :FOR DE=1 TO 150:NEXT
    :POKE 54276,0:RETURN
9810 :
9820 REM S/R TO REORGANISE
    ARRAYS TO MAXIMIZE MEMOR
    Y STORAGE
9980 DIM A$(FD,200):RC=1
    :AT=52038:RETURN
```

PROGRAM: MF64.FORM

```
0 REM[SPC2]*****************
    *******
10 REM *[SPC6]MICROFILE64
    [SPC5]*
20 REM *[SPC4]FORMAT PROGRAM
    [SPC4]*
30 REM *VER1.2(C)1985 G.THOM
    AS*
40 REM *********************
    ***
64 :
65 REM SET UP SPRITE CURSOR
```

```
70 AT=52038:V=53248
   :FOR L=832 TO 896:POKE L,0
   :NEXT:FOR L=832 TO 854 ST
   EP 3:POKE L,255:NEXT
80 REM SET SPRITE POINTERS
90 PRINT"[CLEAR]"
   :POKE 2040,13:POKE V,24
   :POKE V+1,58:POKE V+21,0
   :POKE V+39,1
100 :
110 POKE 19,0:SYS AT,4,8
   :INPUT"[YELLOW,SH]OW MANY
    FIELDS(1-9)";FD
   :IF FD<1 OR FD>9 THEN 110
120 FD=FD-1
130 REM SWITCH ON INPUT INTE
   RRUPT
140 SYS 51854:PRINT"[CLEAR,
   RVSON,WHITE]F1-[SE]XIT[C7,
   SPC,WHITE]F3-[SC]LEAR[C7,
   SPC,WHITE]F5-[SN]EXT[C7,
   SPC,WHITE,SR,SE,ST,SU,SR,
   SN,C7,SPC]TO STORE[WHITE]
150 SYS AT,0,24:PRINT"......
   ........................
   ........[HOME,DOWN]"
   :POKE 2023,46
160 POKE 198,0:POKE 254,0
170 POKE 56295,1:POKE 19,255
   :INPUT Q$:SYS 51867
   :P=PEEK(254):IF P=0 THEN
   220
180 ON P-3 GOTO 90,140,190
190 SYS 52156
200 :
210 REM STORE SCREEN
220 SYS 52083:PRINT"[HOME,C7,
   RVSON,SPC,WHITE]F1-[SE]
   XIT[C7,SPC3,WHITE]F3-[SC]
   LEAR[C7,SPC3,WHITE]F5-[SB]
   ACK[C7]:[SPC,SF]IELD 1
   [SPC2]"
230 GOSUB 760:I=-1:X=24:Y=58
   :POKE V+16,0:POKE V,X
   :POKE V+1,Y:GOSUB 1190
240 POKE V+21,1:I=I+1
   :POKE 198,0:GOSUB 1220
   :GOSUB 1190
250 IF G$="[F1]"THEN POKE V+
   21,0:RUN
260 IF G$="[F3]"THEN 220
270 IF G$="[F5]"AND I THEN I
   =I-2
280 IF G$="[F5]"THEN SYS AT,
   31,0:PRINT"[RVSON,C7,SF]
   IELD"I+2:GOTO 240
290 IF(Y/8)-6<0 OR(X/8)-3<0
   THEN I=I-1:4600
300 B$(0,I)=STR$(Y/8)
   :B$(1,I)=STR$(X/8)
310 GOSUB 400:SYS 52156
   :IF G$="E"THEN 240
320 SYS 52038,31,0
   :PRINT"[RVSON,C7,SF]
   IELD"I+2
330 IF I<>FD THEN 240
340 POKE V+21,0:GOTO 1020
350 PS=((X/8)-13)+40*((Y/8)-
   6)
360 POKE 1024+PS,
   128+ASC(MID$(STR$(I+1),2,
   1))
370 POKE 55296+PS,255:RETURN
380 :
390 REM OVERLAY MENU
   :REM THE _ IN NEXT LINE
    IS [C@24]
400 POKE V+21,0:POKE 198,0
   :SYS AT,6,3:PRINT"[WHITE,
   RVSON,SPC]_______________
   _________ ":SYS AT,6,4
410 PRINT"[RVSON,CN,SPC]
   FIELD TYPE DEFINITION
   [SPC2,CH]":SYS AT,6,5
   :PRINT"[RVSON,CM,CT24,C6]
420 SYS AT,6,6:PRINT"[RVSON,
   CN,SPC,SC]URSOR UP/DOWN
    CHOOSE[SPC2,C6]"
   :SYS AT,6,7
430 PRINT"[RVSON,CN,SPC7,SE,
   SPC]TO EXIT[SPC8,C6]"
   :SYS AT,6,8:PRINT"[RVSON,
   CM,SPC4,SR,SE,ST,SU,SR,SN,
   SPC]TO SELECT[SPC4,C6]
440 SYS AT,6,9:PRINT"[RVSON,
   CN,CT24,C6]":SYS AT,6,10
450 PRINT"[RVSON,CN,SPC3]1.
   [SPC,SA]LPHABETIC[SPC8,C6]
   ":SYS AT,6,11:PRINT"
   [RVSON,CM,SPC3]2.[SPC,SN]
   UMERIC[SPC11,C6]
460 SYS AT,6,12:PRINT"[RVSON,
   CN,SPC3]3.[SPC,SD]IGIT
   [SPC13,C6]":SYS AT,6,13
470 PRINT"[RVSON,CN,SPC3,
   BLUE]4[WHITE].[SPC,SA]
   NYTHING[SPC10,C6]"
   :SYS AT,6,14:PRINT"[RVSON,
   CM,SPC3]5.[SPC,SB]ETWEEN
   [SPC11,C6]
480 SYS AT,6,15:PRINT"[RVSON,
   CN,SPC3]6. <[SPC17,C6]"
   :SYS AT,6,16
490 PRINT"[RVSON,CN,SPC3]7.
    >[SPC17,C6]":SYS AT,6,17
   :PRINT"[RVSON,CM,SPC3]8.
    <=[SPC16,C6]
500 SYS AT,6,18:PRINT"[RVSON,
   CN,SPC3]9. >=[SPC16,C6]"
   :SYS AT,6,19
510 PRINT"[RVSON,CN,RVSOFF,
   CT24,RVSON,C6]"
   :SYS AT,6,20:PRINT"[RVSON,
   CM,RVSOFF,SPC24,RVSON,C6]
520 SYS AT,6,21:PRINT"[RVSON,
   SPC,CT24,SPC]":T=55826
530 :
540 REM MOVE MENU CURSOR
550 GET G$
560 IF G$="E"THEN I=I-1
   :RETURN
570 IF G$="[UP]"AND T>55706
   THEN CL=1:GOSUB 780:T=T-40
   :CL=6:GOSUB 780
580 IF G$="[DOWN]
   "AND T<56026 THEN CL=1
   :GOSUB 780:T=T+40:CL=6
   :GOSUB 780
590 IF G$<>CHR$(13)THEN 550
600 BR=(T-55666)/40
   :B$(4,I)=RIGHT$(STR$(BR),
   1):IF BR>4 THEN GOSUB 810
620 GOSUB 1190:SYS 52156
   :GOSUB 350
```

If you do not wish to search the records found by the previous search then typing 'c' will clear the records found and allow you to search the whole file again. This must be typed before every search if it is to be of the whole file. The different searches are;

1. * wildcard – this isn't a true wildcard since it only compares the first few letters of a string i.e. typing 1*SA will search field one for all strings starting with SA.
2. ! compare directly – this is self-explanatory it compares directly the whole string with the whole of the field i.e. 2!EDINBURGH will search field two for any records containing EDINBURGH, so in this case, if a field contained EDINBURGH CASTLE, it would be ignored.
3. @ search string – this will search all the data in a field for the occurrence of a string i.e. 2@DE will search field two for any occurrence of DE so JADE EVANS would turn up.
4. 5. and 6. – these are the inequalities and the same applies to all, with less than and greater than it is less than or greater then but NOT equal to. Also the search can be used for alphabetic as well as numeric data i.e.'A'< 'F'.

Next on the main menu is seven – sort. The simplest of all the options. Just enter the field to be sorted and the rest is done automatically. The eighth option is Format which when selected will first ask you if you wish to load program or data. The program is the format program, if 'p' is pressed this will be loaded and auto-run otherwise pressing 'd' will load the transfer file of data 'format' created by the format program. Both these operations will require (if you are using tape) the winding of the tape to the correct position. The program will prompt you through all this. So if you run the format program by selecting 'p', then design your format, and lead the main program back in, you will use 'd' to load the file saved by the format program.

The penultimate option, nine, is new data, this will prepare the program for new data to be loaded or for a completely new file i.e. format and data will be erased. When this is selected you are asked whether you wish to erase data only or data and defintiions, obviously if you select data only the format will remain, allowing you to load in another data created under this format.

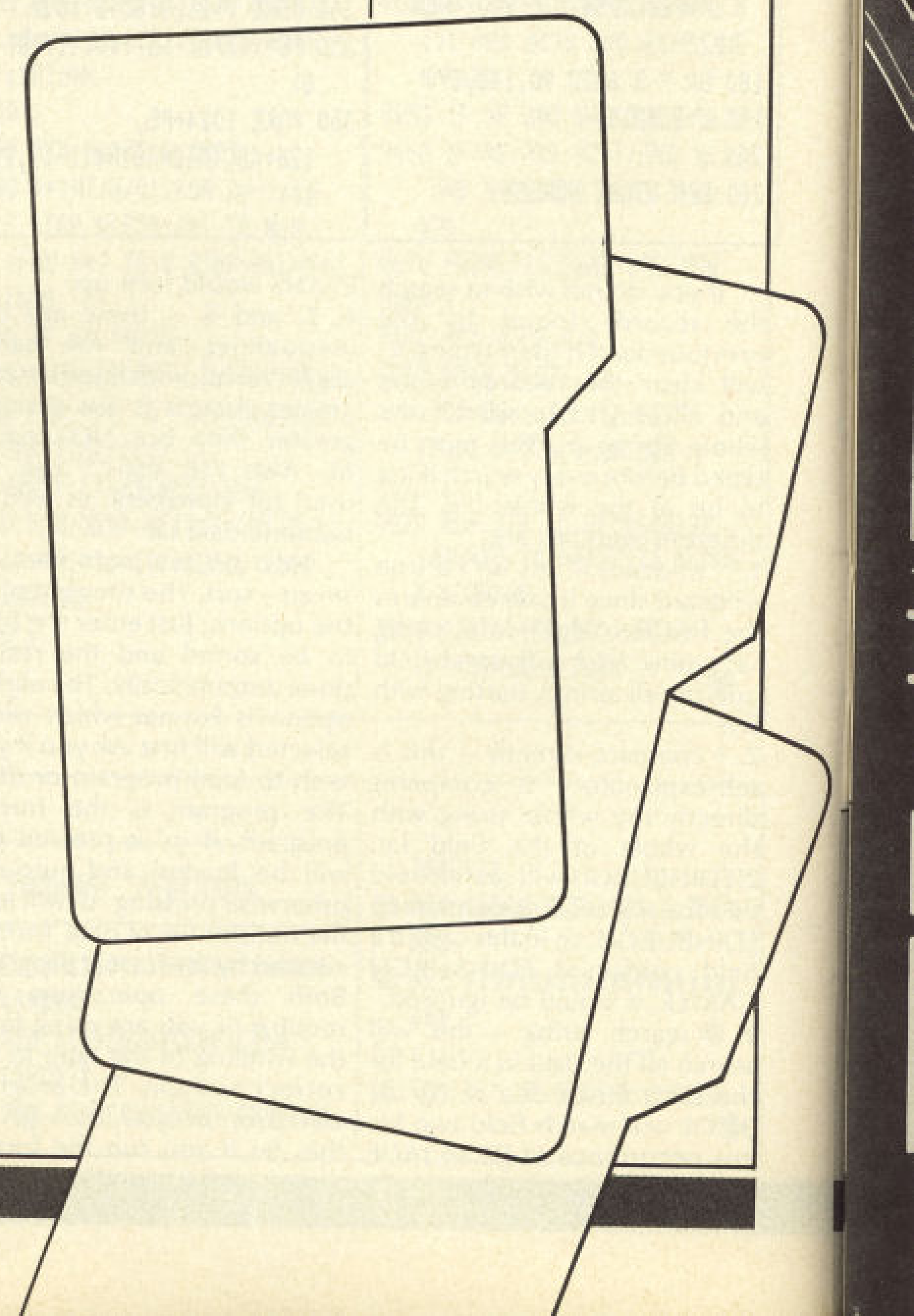
The final option, 0, is to exit the program, this uses the KERNAL routine to completely re-initialise the computer, so be careful.

To finish, I said in the section for the m/c loader, that the max number of records set at 200 could easily be extended to more. The lines that have to altered are 9980 and 1800 – the changes are obvious. If in the event that the program should lock up to an I/O error or an unforeseen bug should crash it and CONT will not continue, then GOTO 30 will return you to the program data intact.

Note

Regarding the use of the format file – the file stored by the format program is only used as a facility to transfer data quickly across to the main program – a copy of the screen is not saved as this takes a lot longer and remains intact above Basic anyway, therefore once you have the format file in memory after using option eight then remember to save a full copy of the format i.e. screen and definition data by using option one selecting save, and pressing 'f' for format when prompted to by the computer. This is a proper copy of the format!

```
630 SYS AT,0,24:PRINT"[GREEN,
 SPC3,SN]UMBER OF CHARACTE
 RS = 1[SPC12,HOME]"
 :POKE 2023,32
640 CO=1:PS=1024+PS
 :POKE PS+1,43:POKE 54273+
 PS,1:POKE 650,128
650 FOR T=0 TO 1 STEP 0
 :GET G$:IF G$=""THEN NEXT
660 IF G$=CHR$(13)THEN B$(3,
 I)=STR$(CO):GOSUB 720:T=1
 :NEXT:GOSUB 1190:RETURN
670 IF G$=CHR$(20)AND CO>1 T
 HEN CO=CO-1:POKE PS+CO,43
 :POKE PS+CO+1,32:GOTO 710
680 IF PEEK(PS+CO+1)<>32 OR
 CO=255 OR PS+CO=1983 THEN
 NEXT
690 POKE PS+CO,46:CO=CO+1
 :POKE PS+CO,43
 :POKE 54272+PS+CO,1
710 SYS AT,25,24:PRINT STR$(
 CO)"[SPC2,HOME]":NEXT
720 EF=1:I2$="":SYS AT,0,24
 :PRINT"[YELLOW,SF]
 IELD"STR$(I+1)" NAME?
 [SPC16,HOME]"
730 POKE 251,0:POKE 252,24
 :POKE 253,4:SYS AT,13,24
 :XP=14:YP=24:GOSUB 960
740 IF I$=""THEN 720
750 AS=ASC(I$):IF AS>64 AND
 AS<91 THEN I$=CHR$(AS+128
 )+RIGHT$(I$,LEN(I$)-1)
760 B$(2,I)=I$:SYS AT,0,24
 :PRINT"[C7,RVSON,SM]OVE
 CURSOR TO INPUT POS &
 PRESS[SPC,WHITE,SR,SE,ST,
 SU,SR,SN,HOME]
770 POKE 2023,160
 :POKE 56295,190:RETURN
780 POKE T,CL:RETURN
790 :
800 REM SET UP VARS FOR M/C
 INPUT
810 K$=B$(4,I):POKE 251,0
 :POKE 252,6:POKE 253,3
 :YP=20
820 ON ASC(K$)-52 GOSUB 830,
 880,900,920,940:RETURN
830 XP=21
840 SYS AT,8,20:PRINT"[SL]
 OWER LIMIT?":SYS AT,20,20
 :GOSUB 960:B$(5,I)=I$
 :POKE 251,0
850 SYS AT,8,20:PRINT"[SU]
 PPER LIMIT?[SPC11,RVSON,
 CG]":SYS AT,20,20
 :GOSUB 960:B$(6,I)=I$
860 IF VAL(B$(6,I))<VAL(B$(5,
 I))THEN 840
870 B$(4,I)="3B":RETURN
880 XP=19:SYS AT,8,20
 :PRINT"[SL]ESS THAN?"
 :SYS AT,18,20:GOSUB 960
 :B$(5,I)=I$
890 B$(4,I)="3<":RETURN
900 XP=22:SYS AT,8,20
 :PRINT"[SG]REATER THAN?"
 :SYS AT,21,20:GOSUB 960
 :B$(5,I)=I$
910 B$(4,I)="3>":RETURN
920 XP=23:SYS AT,8,20
 :PRINT"< THAN OR =TO?"
 :SYS AT,22,20:GOSUB 960
 :B$(5,I)=I$
930 B$(4,I)="3<=":RETURN
940 XP=23:SYS AT,8,20
 :PRINT"> THEN OR =TO?"
 :SYS AT,22,20:GOSUB 960
 :B$(5,I)=I$
950 B$(4,I)="3>=":RETURN
955 REM GRAPHICS CHAR IS CBM
 O [CRP]
960 PRINT"[CO,UP]"
 :SYS AT,XP,YP:I$=""
 :FOR T=0 TO 1 STEP 0
 :SYS 51885:P=PEEK(254)
970 IF P=20 THEN I$=LEFT$(I$,
 PEEK(251)):NEXT
980 IF P<>13 THEN I$=I$+CHR$
 (P):NEXT
990 RETURN
1000 :
1010 REM TRANSFER FORMAT TO
 MAIN PROG
1020 PRINT"[CLEAR]"
 :SYS AT,5,3:PRINT"[SP]
 RESS 'R' TO RE-RUN
1023 SYS AT,3,6:PRINT"[WHITE,
 SS]AVE TO[SPC,ST]APE OR
 [SPC,SD]ISK(T/D)?
1029 GET G$:SYS AT,29,6
 :PRINT G$"[DOWN2]
1031 IF G$="T"THEN PRINT"
 [DOWN,RIGHT3,SP]RESS[SPC,
 RVSON,SR,SE,ST,SU,SR,SN,
 RVSOFF,SPC]TO SAVE"
 :WAIT 197,1:OPEN 2,1,1,
 "FORMAT":GOTO 1070
1035 IF G$="R"THEN RUN
1040 IF G$<>"D"THEN 1029
1050 PRINT TAB(5)"[DOWN3,SS]
 AVING....
1060 OPEN 15,8,15,"IO"
 :OPEN 2,8,2,"@0
 :FORMAT,S,W"
1070 PRINT#2,FD:FOR G=0 TO 6
 :FOR H=0 TO FD
 :IF B$(G,H)=""THEN B$(G,
 H)=" "
1080 PRINT#2,B$(G,H):NEXT
 :NEXT
1090 IF G$="D"THEN CLOSE 15
1100 CLOSE 2
1110 IF G$="T"THEN POKE 198,0
 :GOTO 1140
1120 PRINT"[CLEAR,DOWN2]
 LOAD"CHR$(34)"MF64.MAIN"C
 HR$(34)",8":POKE 198,2
 :POKE 631,13
1130 POKE 632,13:PRINT"
 [DOWN4]RUN[HOME]";:NEW
1140 PRINT"[CLEAR]"
 :SYS AT,2,6:PRINT"[SW]IND
 TO MAIN PROGRAM & PRESS
 [SPC,RVSON,SR,SE,ST,SU,SR,
 SN,DOWN2]
1150 GET G$:IF G$<>CHR$(13)T
 HEN 1150
1160 POKE 198,1:POKE 631,131
 :NEW
1170 :
1180 REM MAKE SOUND
1190 POKE 54276,65
 :FOR DE=1 TO 150:NEXT
 :POKE 54276,0:RETURN
1200 :
1210 REM SPRITE MOVE ROUTINE
1220 FOR T=0 TO-1 STEP 0
 :GOSUB 1340:T=(G$=CHR$(13
 )OR(G$>="[F1]"AND G$<="
 [F5]"))
1230 REM ANY 4 DIRECTIONS
1240 IF G$="[RIGHT]
 "AND X<336 THEN X=X+8
1250 IF G$="[LEFT]
 "AND X>24 THEN X=X-8
1260 IF G$="[DOWN]
 "AND Y<232 THEN Y=Y+8
1270 IF G$="[UP]"AND Y>58 TH
 EN Y=Y-8
1280 REM CONTROL SPRITE1 MSB
1290 IF X>255 THEN POKE V+16,
 1:F=1:POKE V,X-255
 :POKE V+1,Y:NEXT:RETURN
1300 IF F THEN F=0
 :POKE V+16,0
1310 :
1320 REM FLASH SPRITE
1330 POKE V,X:POKE V+1,Y:NEXT
 :RETURN
1340 FOR W=0 TO 40:GET G$
 :IF G$<>""THEN RETURN
1350 NEXT:POKE V+21,0
1360 FOR W=0 TO 40:GET G$
 :IF G$<>""THEN POKE V+21,1
 :RETURN
1370 NEXT:POKE V+21,1
 :GOTO 1340
```





Runecaster delves once more into the secrets of adventuring.

AS WE WATCH MANY SOFTWARE houses striving to improve the graphics which now illustrate most adventures, it is interesting to talk to the actual players of these modern-day marvels.

Whilst nearly everyone agrees that swiftly drawn, colourful graphics can always enhance the appearance of an adventure, most of the players I have spoken to recently, also go on to say that unless the pictures have some relevance to the game play, then the contents – either descriptive text and/or good puzzles – is what really matters.

Following this is usually a discussion on the attributes of the latest parser. These too seem to become more complex as time goes by, with longer and longer sentences being understood and actioned with astounding accuracy. But do many players use these wonder-workers? Most, it seems, tend to use the least possible number of words and letters. The exception occurs when repeating a set series of commands to get to a previously attained position.

Interesting ain't it? Who are we all trying to kid? Probably the only people to gain from all this are the promoters of programs – you have to admit it sounds better if you are advertising an all-singing, all-dancing program. Then there is the newcomer to adventures. It certainly provides something to look at while trying desperately to think about what to do next!

Let us know what you think. And give us some examples of what you like listed in the order of preference. Whilst thinking about it, try playing a text only adventure like The Secret of St Brides.

Back to School



St Brides is a real location and a real school. Girls pay money to stay there but it is no ordinary school. It is as though time was stopped 50 years ago. The scenario is straight out of those storybooks our parents used to read, although this adventure takes things a step further.

You play the part of a recently joined 'pupil', puzzled by the apparently total belief of all the others, that the time is 1929 and not 1986. Your self-determined quest is to find out The Secret of St Brides.

The program has been around for some time for the Spectrum but has only recently been produced for the C64. It is a Quill-based game and is not easy! There is plenty of descriptive text and lots of things to find. The problem is working out what to do with them.

There are two approaches to the game – the basic one of tracking down what is going on and also an additional one to find the Amulet. If you can find it and let St Brides know, the school will award you an 'A' Level in Adventuring.

The vocabulary is fairly extensive and the problems met along the way are devious – the instructions suggest that they may all be solved by logic, inventiveness and a little homework. All I can say is that I must have missed some of the lessons where they explained this homework. I'm stuck!

A neat touch is the option to save a game position to either disk or tape (the program is on tape). It is not an expensive program but it will keep you frustrated for hours. Try it and see, then let me know what to do with the cat basket!

Come Home Mike Hammer

The latest detective adventure to hit the C64 is from US Gold's All American Adventure series. It's entitled Masquerade. This is a fairly middle of the road program without unusual features to make it remarkable. It is only available on disk, from which the graphics are called up each time they are needed and take about eight seconds to appear.

The pictures are clear and colourful and are in a semi-cartoon style that comes across well. There are three modes of presentation – text only, all graphics shown, and a neat variation called up by the command 'MIXED'. This will display text only but a null 'RETURN' flips the display into 'all graphics' until the next null 'RETURN'.

You play the part of a private-eye on the trail of a Mr Big in the crime world. The accent is on observation and you must not expect to see everything immediately at first glance. Various messages and descriptions are not repeated so have your notebook handy!

There is a time element to your play, as the initial locations will explode into thin air, 75 minutes (not real time!) after you start out on your investigations. This could be crucial as you must intercept a telephone message before this happens!

The command interpreter is the basic VERB/NOUN input and the vocabulary does not appear to be very large. The first five letters of most commands must be given for them to be actioned. This is very important since the response to anything it does not recognise is "SORRY CAN'T DO THAT RIGHT NOW". Fortunately this does not apply to direction commands (N,S,E,W) and things like inventory (I) and look (L).

The general style is reminiscent of Mindshadow (which is now available on cassette) but does not have quite the same polish exhibited by Activision's game. It should prove an ample challenge to the detective brigade and although not excessively difficult, is probably not to be recommended to the novice adventurer.

Diabolical

I always knew that I had a strong tendency towards the more active type of adventure games – Halls of Death, The Valley, Exodus: Ultima III and more recently the reconstituted Temple of Apshai Trilogy – but I never thought that I would suffer a program that crashed so many times, that I lost count.

Just browsing through the shelves of a software emporium not so far from home base, I came across Telengard produced by Eclipse Software for The Avalon Hill Game Company. I thought I remembered seeing this advertised in some American magazines but could not remember it being pushed at all in this Country, so, at £7.95, I gave it a try.

There have been many attempts to reproduce Dungeons and Dragons style games on computers. Some have been reasonably successful, some have been pretty ghastly. Telengard, I find horribly addictive, but it is true to the main aim of a D & D type of scenario.

Telengard is a graphics adventure in which the main aim is to gain experience and find treasure whilst fighting off all the nasties that hinder your progress. The graphics are reasonable and the tension mounts from the first few moves to the last moments of life.

You start at the bottom of stairs that lead up to a congenial inn. So far so good. All around you is a maze of passages. These you must explore to find treasure, gems, gold, silver, even refuse! You start out with sword, shield and armour, all of a pretty mediocre brand, because if you search diligently you can find better quality items, indicated by such as "A+7 SWORD".

All of this is very much par for the D&D course, even the opening determination of your characteristics is very sensible. You see a series of random numbers allocated for your character's attributes – strength, intelligence, wisdom, constitution, dexterity and charisma.

These series of numbers will continually be updated until you press 'RETURN' to signify your acceptance of the present batch. This may sound as though you can cheat to get the character you want, but in practice, all this means is that you can sway the result by concentrating on one or two characteristics – you are not likely to see an 'all 18s' series very often! You may also choose your character's name.

The instruction booklet firmly recommends that you choose a character with a sound constitution! Heed the advice well, it's based on knowledge, of the game. A character's 'hit points' are initially equal to its constitution. As these are whittled away each time you are hit by a monster, the larger the starting value the better.

Lost 'hit points' may be regained by a restful night's zizz at an inn. Unfortunately, it is not often possible to return alive to said inn in time to recuperate!

Although you start beneath an inn, this is by no means the only inn to be found – there are lots of them. Thank the gods for small mercies! On arriving at an inn, any gold you have with you is banked safely away and your experience credited with the amount of gold with which you arrived.

All valuable treasure – gems, silver etc. – is converted to the gold standard for this calculation. Neither does it matter which inn you decide to visit, they are all linked by 'computer-inn', a fantasy world version of Barclay's Bank!



As your experience increases, certain values trigger a change in your 'hit points', this means that you can take more punishment and can venture further into Telengard in search of more dangerous monsters and greater loot.

Amongst the maze of rooms and caverns which you explore will be found numerous stairways that reach down into the depths of Telengard – travel that way at your peril! It is quite dangerous enough for even the Level three adventurer to travel far from the initial position, let alone look for further trouble down below!

Not only will your level be incremented by your increase in experience but also your ability to cast spells will increase. Initially you only have the skill to cast level one spells, and only one of those between visits to rest and recuperate at one of the thoughtfully provided alehouses.

Magical spells are divided into six levels of relative power. Each spell level has six distinct spells. Characters at experience Level one or two, have only the six spells at level one at their disposal. As characters gain levels of experience the use of higher level (and therefore more powerful) spells becomes possible.

Every three levels of experience gained, permits the use of the next higher level of spells. Spells are of roughly two types – 'combat' spells and 'duration' spells. The former are active offensive spells such as 'magic missile', lightening bolt' or 'fireball'. The latter allow you to cure wounds, search for traps or pass through walls etc.

All commands are given via the keyboard as single key inputs and, except in the novice's training exercise (entered by calling your character Demo!), action takes place in 'real time', so you have to have your wits about you. There is in fact plenty of time to take the appropriate action – but it does not seem like it when you first start!

Commands are divided into two types – 'action' and 'encounter'. There are nine of the first type and they are very easy to remember. 'H' for help will tell you what they are! The 'encounter' commands are 'F'ight, 'E'vade, 'C'ast (a spell) and to pick up something you will be prompted to press 'RETURN'. Movement is controlled by the four keys W,X,F,A (NSEW), U and D.

There are 20 different types of monsters roaming around Telengard. Most of them are hideously opposed to your continuing existence! A very small proportion meet you with the greeting that they "like your body". That can be taken two ways to start with, initially I thought they were going to eat me but, they actually help you, perhaps curing you of all wounds or maybe giving you a powerful weapon.

The monsters have a similar system to you and your experience, so they have a different level, this level and their rank (Gnoll, lowest – Dragon highest), determines how many experience points you gain in defeating them. It also determines how difficult they are to defeat! Although you may be only on the first level, a surprising number of powerful creatures appear right from the start.

When I first came up against a level 60 Giant, I thought there must have been a glitch in the program but then I found the instruction booklet hints at level 76! As befits a game of this type, some of the monsters have quite nasty habits – several drain your energy and knock you down a level of experience. Dragon fire is not much fun either!

There are a number of 'features' such as stairways, pits, altars etc. which remain at the same locations – if you can find them again. Fountains seem to crop up quite often. The water from these varies in colour as does its properties when drunk. Some colours are beneficial, and, surprise, surprise, some are not!

There are potions to heal and also those to increase your strength. Various

artefacts found have 'plus' factors. Careful that you don't inadvertently press 'RETURN' to pick up an item which has a smaller 'plus' factor than the one you already have. Its very easy to do in the heat of the moment! Sadly, you will have lost the one you were carrying.

I have found items up to 'plus 8' but I suspect that there are more powerful creations deeper down. One of the most useful is a 'Ring of Regeneration', this aids the recovery of your hit points by its 'plus factor' with each step you take, a definite necessity for exploration below level two.

There are also the useful 'Scrolls of Rescue' – you may pick up several of these. Using one will take you back to your original place of entry to Telengard (under that inn), but there is a cost, any gold you have with you is lost on the way. This is sometimes a fair price to pay if you have been teleported to an unknown corner of the underworld.

Characters may be saved on a separate cassette and I would recommend doing this immediately you enter the game – especially if you have taken any time in trying to get the character's attributes to your liking!

The instruction booklet burbles on about being able to load dead characters if their name begins with SV – I think this must be a throw back to a disk based predecessor, as I died with monotonous regularity and had no trouble time-shifting my character to a 'save' before he died!

I hope I have whetted a few appetites of the more adventurous amongst you with what has gone before perhaps now is the time to look at the other side of Telengard.

For a start, it cannot be in the true spirit of the dedicated D & D game. Why? Because I do not think you could stay alive for long enough to ever get anywhere without 'saving' your character to tape at very frequent intervals. I was prepared to accept this as part of the 'overall battle', others may not be so sanguine about it.

There are other problems you have to overcome too. And not part of the game either! I have never had a program crash so often before. It loaded perfectly and within the first few moments crashed with a SYNTAX ERROR IN LINE XXXX. Sometimes RUN got it going again, only to crash once more with a syntax error in a different line!

Once past the first few minutes, things were OK until I 'saved' a character. Most times this went perfectly but there were many occasions when the program crashed after a successful 'save' and the main program had to be loaded from scratch (about four minutes).

The program as you must have guessed from the above is written in Basic and of course is a little slow to re-draw your character's surroundings as you move location. But it is not that slow, if it was, I do not think I could have become so involved. In this day and age it is surprising that the present version was not compiled. One must surely ask – why not?

Neither does the keyboard buffer get cleared, so you have to watch that you do not type in ahead of yourself or you will find you have not picked up that '+8 Armour'!

For all these genuine criticisms, I have to admit I enjoyed the game enormously, quite addictive in fact, and that's something, for someone who plays as many adventures as I do. But then I'm a simple barbarian at heart.

I do not think that Eclipse Software still exists – it was an 'imprint' for Longmans (the publishers) but the program is still for sale. The cost is not high, so if you are a frustrated 'dungeon dweller' why not have a try – I really would like to hear what you think of it.

If you cannot find it in your local shops you can get it direct from: Customer Services Dept, Longmans Group Ltd, 4th Avenue, Pinnacles, Harlow, Essex CM19 5AA. The price is £7.95 plus 75p post and packing. Mine may have been a rogue copy, but if yours crashes like mine just consider it as part of the adventure, and please do not complain – you have been warned!

PROFESSIONAL PACKAGING

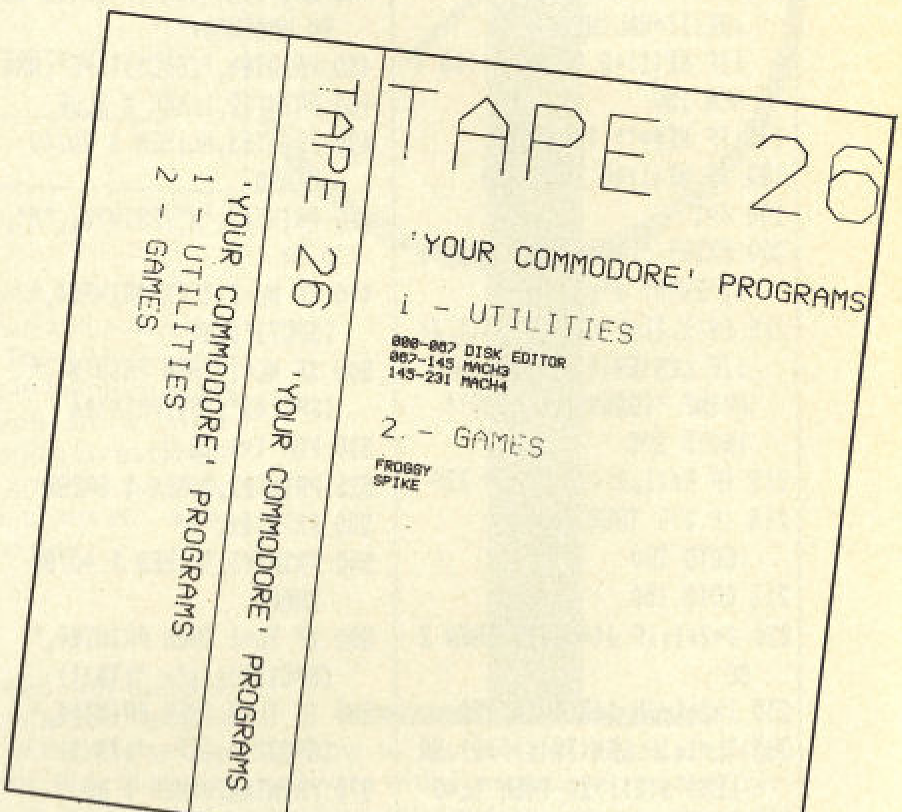
Iain Murray provides a program to smarten up your cassette library.

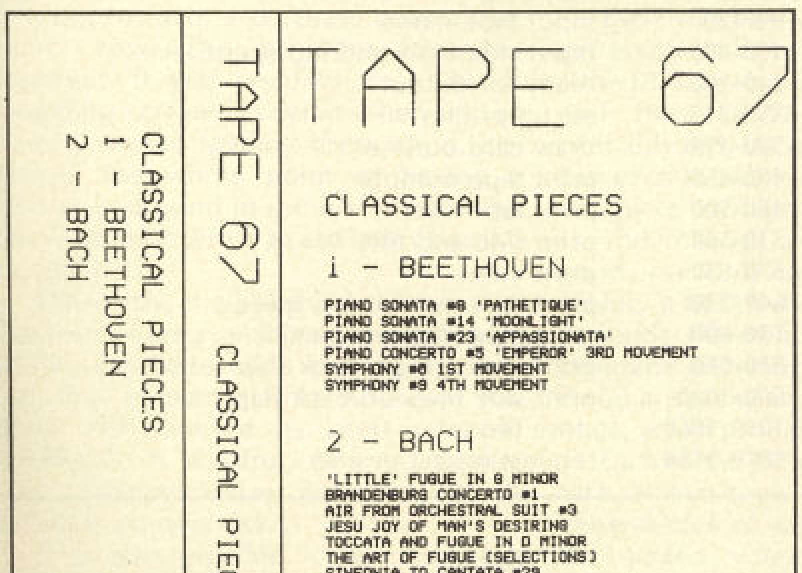
This program allows the user to produce neat inlay cards for cassette tape boxes for music tapes or computer data tapes using a Commodore 1520 Printer/Plotter. It uses many of the features of the 1520 including the four character sizes, four colours, and vertical text for the spine of the inlay.

The program requests input of the tape number, a title for the tape and for each side, and the option to list up to 10 items on the index of each side of the tape. If any input is made incorrectly, then pressing RETURN on its own at the next input will cause a jump back to the previous prompt. The plotter will then produce the required inlay card.

The plotter draws an outline for cutting out and folding the card. The tape number and title are then printed, followed by the title and index for each side. The tape number and title are then printed on the spine. The character size is set automatically depending on the length of the titles required. Finally, the tape title and side titles are printed on the back flap (though if the tape title and side titles are the same, then only one will appear). On completion of the card, the option to print another is given.

Control characters in the text are detailed in preceding REM statements, but these REM statements need not be typed in.





PROGRAM: INLAY MAKER

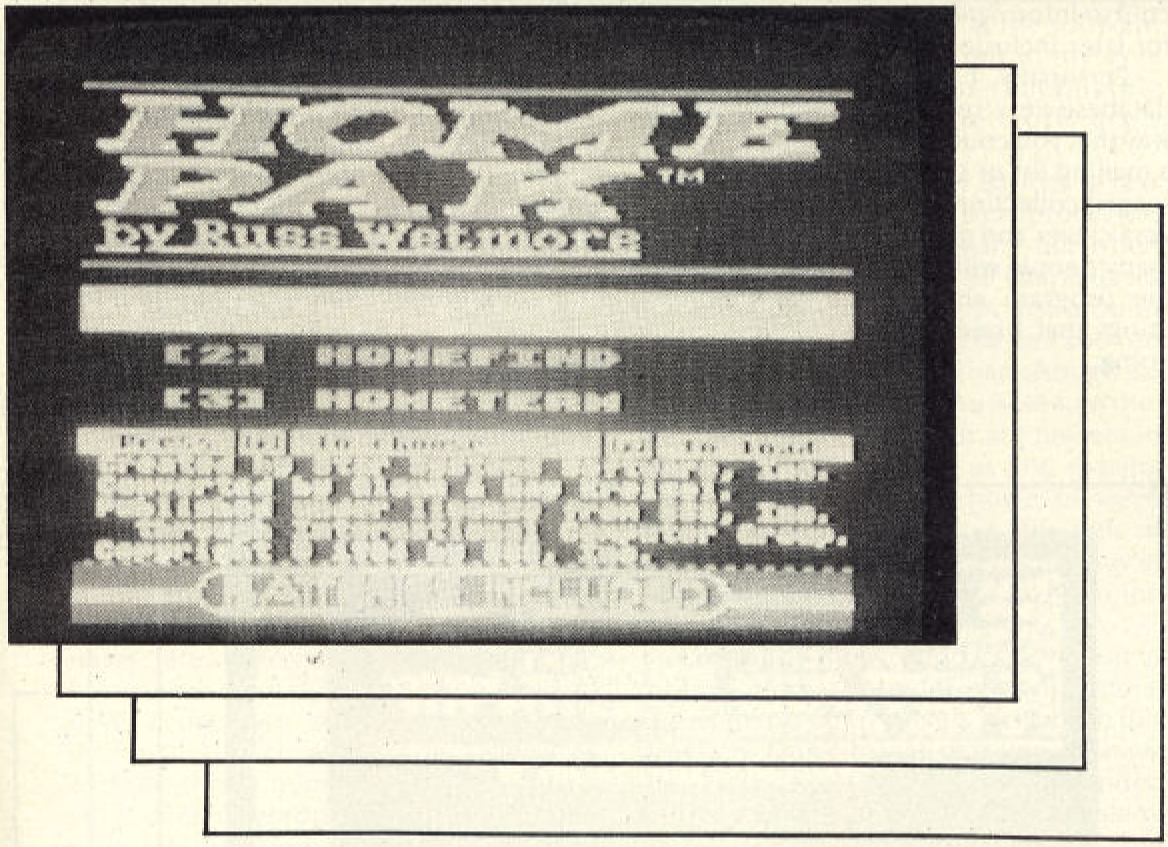
```
10 REM *** TAPE INLAY MAKER
   ***
20 REM *** BY IAIN MURRAY
   (C) 1986 ***
30 REM *** FOR YOUR COMMODOR
   E ***
35 GA$="[SPC55]"
40 DIM E$(2,15)
50 POKE 53280,6:POKE 53281,6
60 PRINT "[CLEAR,WHITE,DOWN3,
   RIGHT13,RVSON,SPC]TAPE
    INLAY MAKER[SPC,RVSOFF]"
70 PRINT "[DOWN3]GIVE TAPE
   NUMBER : ";
80 TN$="":INPUT TN$
   :TN=INT(VAL(TN$))
   :IF TN>99 THEN PRINT "TOO
   BIG!":GOTO 70
85 IF TN$="" THEN 70
90 PRINT "[DOWN2]GIVE TAPE
   NAME : ";
100 TN$="":INPUT TN$
   :IF LEN(TN$)>25 THEN PRIN
   T "[DOWN]TOO LONG!"
   :GOTO 90
105 IF TN$="" THEN 70
110 NL=2:IF LEN(TN$)>12 THEN
   NL=1
```

```
120 I=1
130 PRINT "[DOWN2]GIVE SIDE";
   I;"TITLE : ";
140 T$(I)="":INPUT T$(I)
   :IF LEN(T$(I))>42 THEN PR
   INT "[DOWN]TOO LONG!"
   :GOTO 130
142 IF LEN(T$(I))>0 THEN 150
144 IF I=1 THEN 90
145 IF NE(1)=0 THEN I=1
   :GOTO 130
146 I=1:J=NE(1):GOTO 200
150 PRINT "[DOWN2]HOW MANY
    ENTRIES FOR SIDE";I;
   "INDEX
160 PRINT "(0-10) : ";
170 NE$="":INPUT NE$
   :NE(I)=VAL(NE$)
   :IF NE(I)<0 OR NE(I)>10 T
   HEN 150
175 IF NE$="" THEN 130
180 IF NE(I)=0 THEN 230
190 J=1
200 PRINT "[DOWN]GIVE SIDE";
   I;"ENTRY #";J;" : ";
210 E$(I,J)="":INPUT E$(I,J)
   :IF LEN(E$(I,J))>50 THEN
   PRINT "[DOWN]TOO LONG!"
   :GOTO 200
212 IF E$(I,J)<>"" THEN 220
214 IF J>1 THEN J=J-1
   :GOTO 200
216 GOTO 150
220 J=J+1:IF J<=NE(I) THEN 2
   00
230 I=I+1:IF I<3 THEN 130
235 TL=1:IF LEN(T$(1))>21 OR
    LEN(T$(2))>21 THEN TL=0
240 PRINT "[DOWN6,RIGHT10]
   PRESS[SPC,RVSON,SPC]SPACE
   [SPC,RVSOFF,SPC]TO PRINT"
250 GET A$:IF A$<>CHR$(13) T
   HEN 255
251 IF NE(2)>0 THEN I=2
   :J=NE(2):GOTO 200
252 I=2:GOTO 150
255 IF A$<>" " THEN 250
260 PRINT "[DOWN3,RIGHT14,CA]
   ------------[CS]"
270 PRINT "[RIGHT14,S-,SPC2]
   PRINTING[SPC2,S-]"
280 PRINT "[RIGHT14,CZ]
   ------------[CX]"
290 OPEN 4,6:OPEN 1,6,1
   :OPEN 2,6,2:OPEN 3,6,3
   :OPEN 44,6,4
300 PRINT#2,0:REM * BLACK
310 PRINT#44,0:REM * UPRIGHT
320 PRINT#1,"H":REM * HOME
330 PRINT#1,"D",469,0
   :REM * TOP LINE
340 PRINT#1,"D",469,-505
   :REM * GOTO BOTTOM
350 PRINT#1,"D",0,-505
   :REM * BOTTOM LINE
360 PRINT#1,"M",76,-505
   :REM * GOTO SPINE 1
370 PRINT#1,"D",76,0
   :REM * SPINE 1
380 PRINT#1,"M",139,0
   :REM * GOTO SPINE 2
390 PRINT#1,"D",139,-505
   :REM * SPINE 2
400 PRINT#1,"H":REM * HOME
410 PRINT#3,3:REM * 10 CHARS
420 PRINT#4:REM * LINE FEED
430 PRINT#2,3:REM * RED
440 NM$=STR$(TN):IF TN<10 TH
   EN NM$=NM$+" "
450 PRINT#4,"[SPC3]TAPE";NM$
460 PRINT#2,1:REM * BLUE
470 PRINT#3,NL:REM * 20/40
    CHARS
480 PRINT#1,"H":PRINT#1,"M",
   0,30
490 IF NL=2 THEN PRINT#4,"
   [SPC7]";TN$
500 IF NL=1 THEN PRINT#4,"
   [SPC14]";TN$:PRINT#4
510 FOR I=1 TO 2
520 PRINT#2,2:REM * GREEN
530 PRINT#1,"H"
540 PRINT#3,TL:REM * 40/80
    CHARS
550 IF TL=1 THEN PRINT#4,"
   [SPC13]";I;"- ";T$(I)
560 IF TL=0 THEN PRINT#4,"
   [SPC27]";I;"- ";T$(I)
570 PRINT#3,0:REM * 80 -
S
580 IF NE(I)=0 THEN 625
590 PRINT#2,0:REM * BLACK
600 FOR J=1 TO NE(I)
610 PRINT#4,"[SPC28]";E$(I,J)
620 NEXT J
625 PRINT#4:PRINT#4:PRINT#4
630 NEXT I
640 MB=(10*(NE(1)+NE(2)))+(2
   0*(TL+1))+160
650 SP$="TAPE"+STR$(TN)
   :PS$=" "+TN$
660 PRINT#3,2:REM * 20 CHARS
670 PRINT#2,3:REM * RED
680 PRINT#44,1
690 PRINT#1,"H":PRINT#1,"M",
   0,MB
700 FOR I=1 TO LEN(SP$)
710 PRINT#1,"I":PRINT#1,"R",
   0,18
720 PRINT#4,"[SPC4]
   "+MID$(SP$,I,1)
730 NEXT I
740 PRINT#2,1:REM * BLUE
750 PRINT#3,NL:REM * 20/40
    CHARS
760 FOR I=1 TO LEN(PS$)
770 PRINT#1,"M",0,9*NL
780 IF NL=2 THEN PRINT#4,"
   [SPC4]"+MID$(PS$,I,1)
790 IF NL=1 THEN PRINT#4,"
   [SPC8]"+MID$(PS$,I,1)
800 NEXT I
810 MB=((18*LEN(SP$))+(9*NL*
    LEN(PS$)))+40
820 PRINT#1,"M",0,MB
830 C1=2:C2=1:PRINT#3,1
   :REM * 40 CHARS
840 IF TL=0 THEN C1=1:C2=2
   :PRINT#3,0:REM * 80 CHARS
850 F$=TN$+LEFT$(GA$,
   26-LEN(TN$))
860 G$="1 - "+T$(1)+LEFT$(GA
   $,((C2*22)-LEN(T$(1))))
870 H$="2 - "+T$(2)+LEFT$(GA
   $,((C2*22)-LEN(T$(2))))
880 IF T$(1)<>T$(2) THEN 910
890 G$=RIGHT$(G$,LEN(G$)-4)
   :H$=LEFT$(GA$,C2*22)
900 IF T$(1)=TN$ THEN 990
910 PRINT#2,2:REM * GREEN
920 FOR I=1 TO (22*C2)
930 PR$=MID$(H$,I,
   1)+" "+MID$(G$,I,1)+" "
940 IF C1=1 THEN PR$="[SPC3]
   "+PR$
950 PRINT#4,PR$
960 PRINT#1,"M",0,4*C1
970 NEXT I
980 PRINT#1,"M",0,262
   :IF C1=2 THEN PRINT#1,"M",
   0,273
990 PRINT#3,1:REM * 40 CHARS
1000 PRINT#2,1:REM * BLUE
1010 FOR I=1 TO 26
1020 PRINT#4,"[SPC4]";
   MID$(F$,I,1)
1030 PRINT#1,"M",0,8
1040 NEXT I
1050 PRINT#44,0
1060 PRINT#4:PRINT#4:PRINT#4
   :PRINT#4:PRINT#4:PRINT#4
   :PRINT#4
1070 PRINT "[DOWN3,RIGHT12]
   PRINTING COMPLETE"
1080 PRINT "[DOWN3]ANOTHER
    INLAY CARD (Y/N) ?"
1090 GET A$:IF A$="Y" THEN C
   LR:GOTO 30
1100 IF A$<>"N" THEN 1090
1110 END
```

How it Works

10-60	set up and title
70-85	input tape number
90-120	input tape name
130-235	input side titles and index entries
240-280	ready for output
290-310	set up printer
320-390	draw card outline
400-450	print tape number
460-500	print tape title
510-560	print side and title
570-630	print index
640-730	print tape number on spine
740-800	print tape title on spine
810-890	print tape title on back flap
900-1040	print side titles on back flap
1050-1060	close file
1070-1110	request repeat or end

3 into 1



Stuart Cooke puts Ariolasoft's Homepak to work.

IF, LIKE ME, YOU USE A COMPUTER FOR a lot of your work it is very important that it gives quick access to everything that you wish to do. For example you may do a lot of wordprocessing and require access to a database or a spreadsheet occasionally. No problem, I hear you say, buy one of each program. This is a great idea, but one major thing is being forgotten, time.

The C64 and its disk drive are not exactly well known for their speed, in fact most people moan about the lack of it. A typical wordprocessor will take about five minutes to load, a database around the same. Now the problem becomes apparent. If you need to do a lot of swapping between programs then a lot of time is wasted loading them all in, defeating the whole point of having a computer around in the first place. Why use a database when a card index box is a lot quicker?

Obviously, if all of the programs that you require are available on one disk, a lot of time can be saved in exchanging disks etc. This is exactly what Ariolasoft has done with one of its latest releases, Homepak. A wordprocessor, database, and communications program are all available on one disk.

It is also possible to go one step further. Wouldn't it be great if all of the programs that you needed to use regularly could all be in the computer's memory at the same time? Press a couple of keys and the program needed would burst into life ready to obey your every command. Team-Mate, a program that looks extremely similar to the 3+1 software that is found on a Plus/4, offers just this facility. With Team-Mate up to three programs can be in memory at any one time. The programs are a Wordprocessor, a Database and a Spreadsheet. As an added bonus a graphics package, for drawing graphs, pie charts etc. is also present on the software disk.

Homepak

As previously mentioned this suite of programs goes some way to solving some of the speed problems of the C64 as all of the programs are on one disk. However they are all quite slow in loading and a great deal of disk swapping is necessary if you need to use the other programs.

Each of the available programs are extremely well presented and easy to use – the 61 page manual makes sure of that – and have facilities that you would probably only expect to see on individual pieces of software costing as much as this complete package.

The manual, even though it is very good, can only be described as microscopic. It has been reduced so that it will fit inside the standard disk box that the programs come in. Get a magnifying glass if you are going to be reading a lot of it at once, you'll probably need it.

Each of the programs are dealt with in turn. Screen shots are used to give you a general idea of what you should see on the screen when certain menus are accessed. And a handy crib sheet at the end of each program's documentation gives a handy reminder of the keys needed to operate the software. I must admit that I fail to see the relevance of a very large section of the manual (seven pages) being given over to an explanation of how to use the telecommunications software with Compuserve and the Commodore Information Service, these are American software services. Come on Ariolasoft, you've gone to the trouble of printing your name on the front of the manual, why not alter the last section so that it refers to a British system such as one of the many bulletin boards available or even Telecom Gold. I wouldn't have thought that too many people would be phoning America so that they can follow your instructions.

Hometext – yes you've guessed it – the wordprocessor, has some extremely interesting features. All of the available commands are selected from 'pull down menus' that are controlled by the function keys. This means that when you press the relevant key a menu, for example the printer format menu, will appear on the screen on top of your text, replacing the text underneath when you have finished using the menu. This is great as you never have to memorise any of the commands, such as those for headers or setting margins, as they can all be called up on screen. There is one slight gripe here however. A reminder could have been put on the editing screen so that you

could see at a glance what function key brought up which menu, it's frustrating to go through them all every time you want to do something. I suppose if you were really bothered you could always stick a bit of paper over your function keys.

When you have finished typing your latest novel you can have a look at what the page layout looks like with the view function. This 'draws a picture' of every page with each letter being represented by a dot. This does come in very useful when things need to be positioned correctly, it may even help you to spot your mistakes in the layout.

Of course all of the normal printer facilities such as underlining and spacing are catered for, though headers and footers are dealt with in a strange way. Not only do you have to tell Hometext where a heading starts you must also tell it where the heading finishes. This means that it is possible to have headers that run over more than one line of the paper when printed. I must admit this did leave me a little confused at first as I didn't tell the program where my header finished the first time that I tried to use this function. The view option showed that something was amiss and I was able to correct the problem before I sent the document to the printer, I told you that view was handy.

Homefind – the database – is a little strange. In case you have never used a database I should explain how you would normally use one. Your computer is treated as an electronic card index box. You would set up a series of fields into which you should enter information. You can then ask the computer to find specific information from what it has stored on disk. An example of a layout for a database may be:

NAME
ADDRESS
TELEPHONE

You can no doubt see where the similarity to the old card-index comes in. Well, Homefind is totally different. Yes, it is still used for information storage and retrieval but there is no fixed format as to what can be entered into the system. For example a few entries to the database may be:

Fred's Birthday's August 23rd
Jim's Birthday's June 5th
Fred's Address's 123 Main Street

As you can see you almost talk to the computer, and any information can be stored. Once the information has been stored you can ask questions such as:

What's Fred's Address?

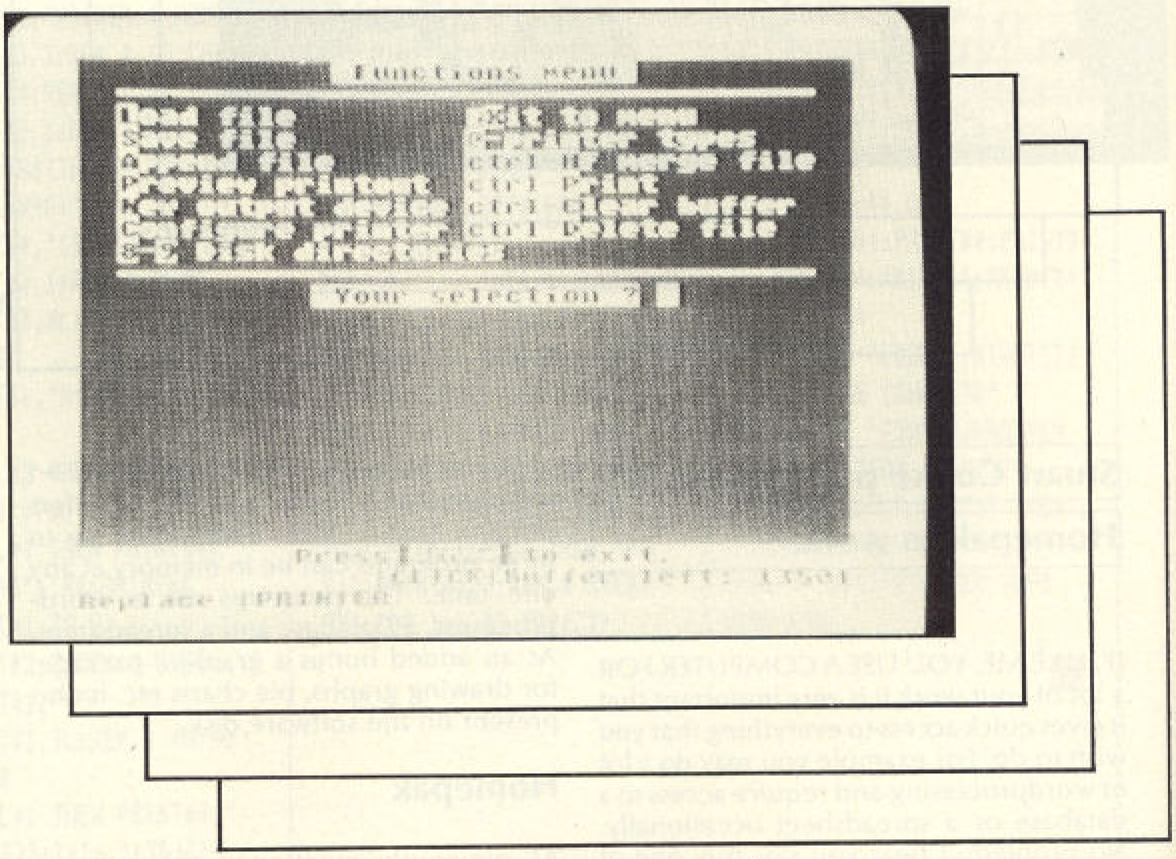
And the answer will appear as if by magic.

There is no provision within the program for getting printouts of specific information though it is possible to keep a printed copy of any 'conversations' that you have with the computer. And of course information can be stored on disk for later inclusion in the wordprocessor.

Personally, I find this structure for a database very restricting, I can't see any way that you could use the program to run a mailing list or store information about a record collection. Even so the program is very clever and great fun to use. No doubt many people will love the 'friendliness' of the program and use it for just those things that I said I couldn't see a way of doing.

One handy facility that the program has is that of Macros. It is possible to set up a file that holds information such as your name, and your password. These macros can then be used to send the information required to the computer saving you a lot of typing if you access a particular system a lot.

Not much more can be said about these packages. They are all very good and would be worth a look at if the asking price was just for one of them. As I have



Hometerm I consider to be one of the most important things about this program. Modems are becoming cheaper all of the time and more and more people are becoming interested in communicating via computer. Before I go any further it is worth stressing that this program is a terminal emulation program which can be used to access bulletin boards and systems such as Telecom Gold. It is not a Viewdata type system and cannot be used with systems such as Prestel or Compunet.

With Hometerm and an RS232 modem it is possible to talk to your friends who have this software and modems, swap programs with each other and talk to each other via your computer keyboard. Many of the bulletin boards now have Commodore sections on them so you can ask questions to other people who use the system etc. In fact the day of the electronic office is here. One person can write an article using Hometext. Send it to someone else via Hometerm and they can then print it out using Hometext.

For the more technical bods out there the following data formats are catered for, ASCII, CBM (commodore characters), Vidtex and the very popular Xmodem.

said I found Homefind a little limiting but no doubt others will love it. All of the programs are well presented and easy to use. If you need any of the programs then the package is well worth the price even though some of the 'polish' may be missing that its more expensive, individual program, competitors have.

All in one

The other package mentioned is Team-Mate. What makes this program stand out from the rest is the fact that it is possible to have all of the programs in memory at any one time. Obviously, this means that you are limited to how much space is available at once for a specific task. Team-Mate gets around this in a very clever way. It allows you to choose, upon loading the software, exactly what you want in memory. Below is the menu presented when you load the software:

(1) Desktop (3 prgs)
(2) Plus Graph (1 prg)
(3) Write File (2 prgs)
(4) Home Office (2 prgs)
(5) Utilities

Dealing with each option in term. Desktop comprises of a Wordprocessor, Spread sheet and file manager (database) all of which are loaded into memory at the same time and very easy to switch between.

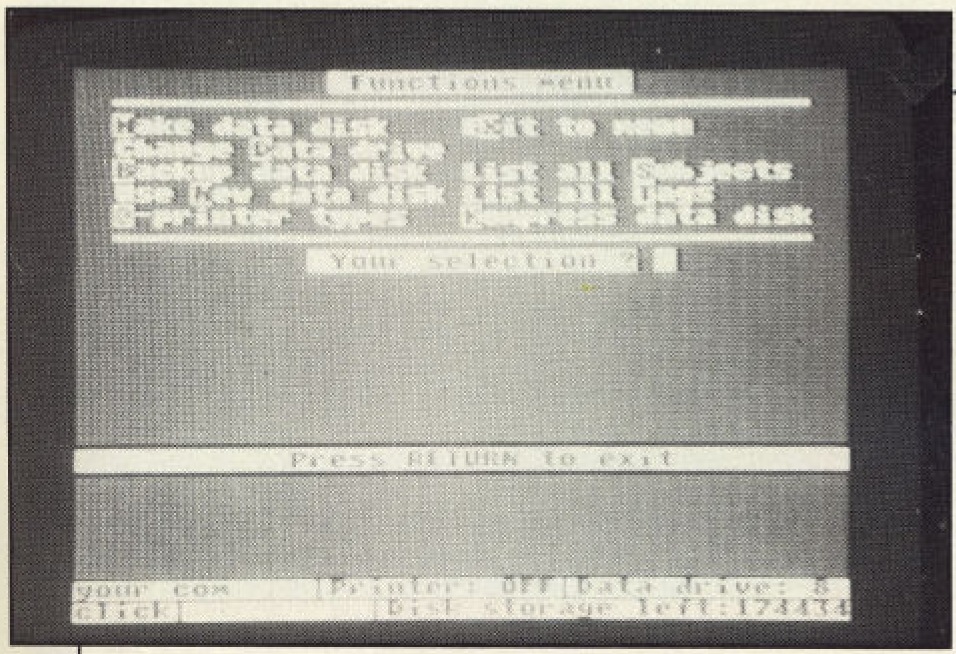
Plus Graph is a stand alone business graphics program that is used to display information from either the spreadsheet or entered by hand.

Write file comprises of the wordprocessor and the file manager both now with a help facility.

Home Office is Wordprocessor plus help and a spreadsheet plus help.

Utilities are such things as 'Format Disk' and 'Rename File'.

As can be seen from the above breakdown quite a number of differing tastes are catered for. My personal favourite is Writefile. This allows me to have a database at my fingertips with information such as company addresses and telephone numbers, and access to a fairly decent wordprocessor at the same time. Up until now the only that this has been possible was to have two C64s on my desk.



As I have previously mentioned the software is very similar to that found in the Commodore Plus/4 computer. The layout of the programs and the instructions for use are very similar, i.e. the same. For this reason the software will probably get the same stick that the Plus/4 did when it arrived on the market, 'only 99 lines in the wordprocessor' and other such comments abounded when that beastie hit the streets. Granted that the software does have its limitations but in my eyes these are more than adequately overcome due to the software's convenience.

As with the Plus/4 only 99 lines of text can be entered into the wordprocessor. This doesn't sound too many but when you realise that a line is 80 characters a quick calculation will show that around 1000 words can be entered before you run out of room. This is more than enough for the standard letter that you wish to write. All of the usual commands are present in the wordprocessor, such as line spacing and margins. There are however some notable omissions, such as the lack of headers and footers. An interesting way of overcoming this problem is given in the manual, but more of this later.

Obviously the wordprocessor is not as sophisticated as many of its competitors. There are no fancy menus or icons in this program. Don't forget that you do have the Help function in the expanded versions which soon solves this problem.

It is remarkably easy to transfer data from both the spreadsheet and database into the wordprocessor. In fact the only way to print information from the database is via the wordprocessor. It is even possible to select certain fields for printing, this makes the program very good for addressing letters or printing labels. It is this facility that allows you to add headers to articles. Simply leave room at the top of every page for the header when you print your text. Then set up your header as a database file, then put your paper back in the printer and tell the program to print the information in the database at the top of every piece of paper, clever huh!

As with Homepak a preview function is also available within the wordprocessor (this function is not present in Desktop) this prints out the text to the screen as it will appear on the printer. The 40 columns of the screen act as a window over the larger 80 columns of the text.

The database or file manager is more of what I would call a real database. Before you use the program you must set up a specific format for all entries, such as the one given earlier in this article. Information is then entered as requested by the program and stored on a disk for retrival at a later date. Again no fancy icons or prompts in this program and it is a little awkward to use in places. But it does its job and is very handy.

Commands available allow the user to move to specified records, search for a specific piece of information, review records, update records and even copy records. It is possible to sort records using a specified field, it is even possible to do a sort on disk with up to three fields.

The expanded file manager, available from write file but without the wordprocessor present, offers even more facilities for the more advanced user. Examples of added commands are 'makekeyfield' and 'showkey' which make a specified field the main field in a record, speeding up all searching and sorting and print the contents of the key field respectively.

The Spreadsheet is not exceptionally large, 50 rows by 17 columns. This means that it is not suitable for use in a large business but is great for working out budgets or totalling expenses. Don't forget you can even get the wordprocessor to print out a letter taking information from the spreadsheet making printing bills very easy. In fact one very nice touch is the ability to have half of the screen displaying the contents of the wordprocessor and the other half the spreadsheet. This makes it very easy for you to see exactly what you are doing.

The manual for the suite of programs is written in such a way that even a beginner could get started without too much difficulty. All aspects of the programs are dealt with via little examples, for instance the section on the spread sheet shows how you could set up a budget sheet showing all the money that you have spent or saved.

Team-mate can only be described as the program that a lot of C64 owners have been waiting for. OK, so it has its limitations but there are ways to get around these. The fact that the programs you are going to need can be loaded into memory just the once and that data can be stored on one disk is superb. Load in the programs in the morning and that's it. Everything is at your finger tips. Now I've started using the programs I wouldn't be without them.

PSI 5 TRADING CO.

US Gold £14.95 Disk

Originality 10 Playability 9 Graphics 7 Value For Money 9



HAVING WATCHED COUNTless episodes of Star Trek, I always thought that Captain Kirk had things very easy commanding a space ship. Especially if you have a reliable crew to carry out your every order quickly and efficiently. Psi 5 Trading Co., from US Gold seemed to be just the game I needed to prove my point. All I had to do was to select a suitable crew to enable me to deliver a cargo to some remote outpost of the universe and collect a handsome bonus. One hour into the game and I was a total wreck. My crew were tearing their hair out in despair and countermanding my orders, looters were stealing my cargo willy-nilly and I no longer knew whether I was coming or going.

Your first decisions are involved with crew selection and the success of your mission could well be put into jeopardy straight away if you get it wrong. There are five positions to be filled in the scanning, weapons, navigation, engineering and repair departments with six candidates for each job. A screen illustrating the candidates is displayed – they may be humanoids, aliens or vaccdroids and you can call up details of each applicant before making your selection. From these resumés, you can learn about a character's qualifications, education, experience, strengths and weaknesses. For example, you may learn that whilst someone knows their job inside out, they fall to pieces under pressure, whilst a rival might be a loner and a poor communicator but won't bat an eyelid when the ship is under heavy attack.

Having agonised over your recruitment, it is on to the mission itself. The screen depicts your communication console with your current view displayed top left and the appropriate crew member top right. The bottom half of the screen gives various status reports whilst the central bars contain assorted indicators for you to monitor and a menu of your current choices.

Your initial menu allows you to read all the pending messages or contact a specific department. If you don't keep in touch with sections regularly, they will go off and do their own thing. Most of your decisions involve what the Americans call prioritizing – working out which orders take precedence. Is it more important to get the ship travelling at full speed or should you divert some power into defensive shields? Which items should be repaired first? Strangely, every department thinks that they should have priority. As things get worse, so the animated pictures of the crew show their feelings – totally laid back or gibbering wrecks. To give some idea of what you have to control, here is a quick look at the five departments and their various tasks.

The scanning department is in charge of the radars and will try and identify and track other spacecraft. In time, they can determine whether a craft is friend or foe, lock on to it and recommend which weapon is best to use against it.

Once an enemy has been identified, the weapons section can attack it with missiles, blasters, cannons and thermos. By analysing various statistics, you can assess how efficient your crew is with each weapon type.

The navigation section will estimate the time of arrival at your destination and show you the risk involved with various routes. You can change speed and take evasive action if necessary.

All sections of the ship have various power requirements and allocating that power is the responsibility of the engineering department.

It will not be long before your ship suffers damage and it is necessary to request the services of the repair section. Items can be operable (at a higher power cost), repairable or destroyed. There are several droids at your disposal and again, you will need to decide what needs to be done, who's available to do it and how long it will take.

Control of the game is very simple via either joystick or keyboard – it is just the decisions that are difficult. There is so much going on, that it will take some considerable time to get the hang of things. Graphically, the game is stunning so you can always sit back and watch as your ship is destroyed around you. A highly original and thoroughly excellent game.

G.R.H.

TOM THUMB

Anirog C-16, Plus/4

Originality 8 Playability 9 Graphics 8 Value For Money 9



YOU, TAKING ON THE PART of that childhood hero Tom Thumb, have entered the tomb of the Pharoah Manilo in order to find the treasure that was buried with him.

Evidently you know where the treasure is, at least the instructions say that you do, but in order to gain access to the treasure you must collect a number of keys that are scattered around no fewer than 178 screens. Yes, that's right, somehow Anirog has managed to get a 178 different screens into the C-16.

You would expect that the screens would be very small in order to fit so many into the game. Well, they aren't.

Tom always stays in the same position on the screen. Whenever he moves around it is the background that scrolls bringing new sections of the playing area into view. An excellent bit of programming when you consider how simple many C-16 & Plus/4 games are.

Tom's journey around the tomb is hindered by all sorts of nasties. There are guards moving their spears up and down which Tom must jump over – just like the arcade game Hunchback – monsters, spiders, snakes and collapsing floors. Tom certainly has his work cut out, even the flowers and plants scattered around are out to get him.

If you fancy a little adventuring then why not buy a copy and give Tom a hand?

S.C.

PHANTOMS OF THE ASTEROID

Mastertronic £1.99 C64

Originality 5 · Playability 7 · Graphics 8 · Value For Money 9



HELPLESSLY FLOATING Around the eerie tunnels inside a great hunk of rock in the sky you are mercilessly assaulted by weird looking bug-eyed monsters whose immaterial qualities allow them to pass through rock and appear when you are least expecting it.

You, a lone astronaut, are armed only with a pathetic laser gun which luckily has an inexhaustible power supply. Ghastly music pings in your ears as you boost yourself through the maze of caves with the aid of your trusty jet pack.

Suddenly, you hit a laser force field and death comes quickly as your whole being is pulverised and you disappear in a puff of sulphurous bubbles.

Then comes the tedious bit. Since you are a normal homo sapien, you possess only one life and must wait for the Game Over screen to conclude several dischordant bars of 'music' before you can send another unfortunate spacer to certain death.

The green force fields aren't too bad since they disappear every few seconds and you can nip through. Apparently there is away to deactivate the blue and purple ones but I never lived long enough to find out how to do it.

This is definitely one of Mastertronic's better offerings. Spend £1.99 and die as many times as you like.

M.C.

MUGSY'S REVENGE

Melbourne House C64

Originality 1 · Playability 1 · Graphics 5 · Value For Money 2



AS THE NIGHT CLOSED IN, black as a Mickey Finn dream, my hands fumbled with the cellophane and loaded the Mugsy file in the sleepy blue glow of the monitor. Colours flashed and I was transported back to the heady, dry days of 1917. As Mugsy blinked under the unaccustomed daylight outside the slammer, the Feds were clamping down on liquor. Through half-closed eyes the Boss dreamed of empire as my spirit slipped easily into his mind.

First we got the boys together and made a deal with the Canadian hoods across the Parallel. All the gin joints had been driven underground like worms in a drought and easy pickings were the harvest. All we needed were the booze, the speakeasies, the dames and the dough.

New Year brings resolutions, mine is to kick the rotgut for the duration and show a profit. I've hired the Pixel Brothers to record the high points in full colour. They even came up with a slick cartoon of 'Chalk' McHugh getting his card marked down the Pool Hall. I'll show it each New Year as a warning to any other creeps till they get bored with the message. Life is still as dull as the shine on felt spats. Even the shoot-outs are rituals of emptiness. Everything is the greyness behind a rainbow, a pig in a poke.

J.G.

BONGO

Anirog C-16, Plus/4

Originality 4 · Playability 7 · Graphics 8 · Value For Money 9



THIS VARIATION OF A PLATform and ladders game finds our hero trying to rescue a damsel in distress. Before the damsel can be rescued Bongo must collect 10 sparkling jewels (flashing dots) from around the screen.

Travel around the playing area is via a number of slides, trampolines, ladders and teleportation chambers. Of course, movement around the platforms isn't easy as the ever-present baddie is out to stop you.

As well as collecting the jewels, Bongo can also get himself extra points by gathering the letters that are floating around the screen.

Ok, so there's nothing original so far. Even so the game has some interesting aspects. For a start all of the characters are extremely large and well defined, the colour facilities of the C-16 and Plus/4 being used to their full. This certainly makes a change from being chased around a screen by a single colour, one-character-high monster.

Secondly, Bongo comes complete with a built-in screen designer. Once you get fed up of running around the screens provided you have the option of designing your own.

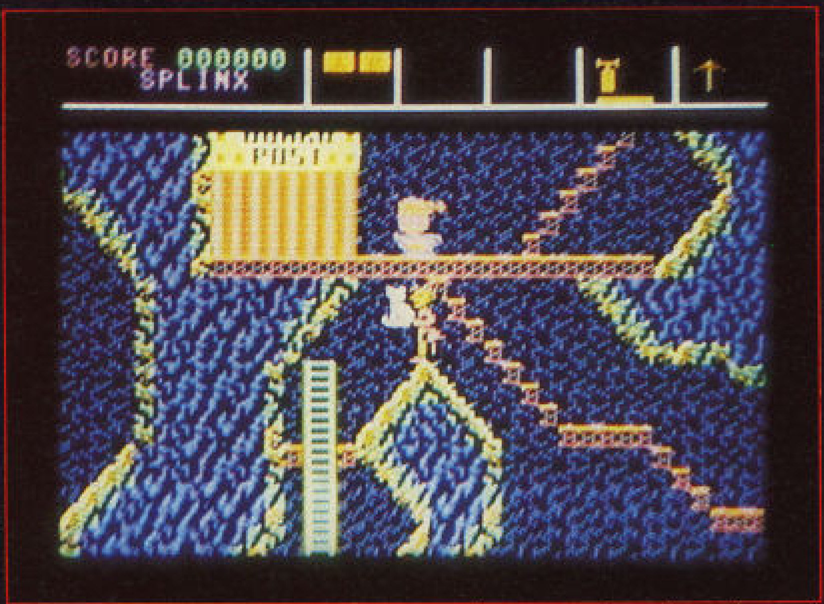
Well done, Anirog! A program that all games-playing C-16 and Plus/4 owners should not be without.

S.C.

DR WHO AND THE MINES OF TERROR

Micropower C64

Originality	Playability	Graphics	Value For Money
5	6	8	8



CLOSELY GUARDED COPYrights had to be skirted around in the production of this game, based around television's most successful science fiction series. The BBC seem pleased to be associated with the product but I suspect that the series' creator decided that it was against the national interest to tarry with Micropower. No doubt royalty fees lie at the heart of this situation.

The result is reflected in the appearance of, dare I say it, Dalek-like creatures called Controllers who guard the TIRU machine which the Doctor must disable before the fabric of the time-space continuum is torn to shreds.

The TIRU (Time Instant Replay Unit) is a temporal editing suite. An instant in time can be recorded, edited and replaced for good or evil purposes by the use of this machine. It is therefore the real time equivalent of our humble VHS video system. The source of its power lies in rods of Heatonite crystal which is only found in sufficient quantity on the planet Rijar. For obvious reasons this is where the TIRU has been built and it is now controlled by the Doctor's arch-enemy, the Master.

The pre-publicity claims that the game starts with the familiar Dr Who theme. Well, if this is true then I'm a toreador. Maybe, there were more copyright problems but judging by the voicing of the existing tune it appears that we haven't missed much. I strongly advise that you play this game with the volume turned off.

The good old Police Box Tardis gradually materialises at the start of the game and out pops the Doctor and his new assistant Spinx, the robotic cat. This white feline is an amazing invention of the Time Lords and it can wander at will through the ensuing unobserved by all except the Doctor. It is a secret weapon in every sense of the word and if lost it will be impossible for the good Doctor to complete his mission to destroy the TIRU and recover the plans.

Splinx can be programmed to perform a range of tasks as long as it does not involve climbing! Whether it suffers from vertigo or not is not explained but the only way to persuade it to go up a ladder is to force the Doctor to carry it. It is nonetheless a very useful beast which can be sent to recover any item which is near one of the markers which the Doctor can throw around the place.

It is not long before Splinx's assistance is necessary. After a short exploratory walk around the Heatonite mine a dinosaurlike creature, a Madrag, will be encountered. Normally this will first be in the form of Madrag eggs which hatch spontaneously as soon as a suitable food source comes near. Guess Who is suitable!

The eggs are also jealously guarded by mother Madrag and this is a useful key to the solution of the first of many problems which must be faced.

The documentation with the game is superb, an area in which Micropower usually excells. Apart from the usual loading and playing instructions the pack also includes a sealed solution to the Madrag problem, a detailed breakdown of the main elements of the game, a map of part of the mining complex and a printout of the major objects encountered in the game. The final card in the pack is vital because it gives the code used to recover the memory capsule containing the TIRU plans. By far the greatest challenge is an encounter with a controller. They are fast and persistent in their pursuit of an enemy. As long as they can see which way you went they will give chase. Fortunately, they cannot climb ladders or move off the grid floor which supplied their power.

The Doctor regenerates each time he is killed. This means that he returns to the TARDIS or the last Cryogenic Sleep Chamber (CSC) that he passed. All of the objects which have been found and tucked away in his copious pockets will be retained but he will always have to relocate Splinx. The CSCs have a second, important purpose. By standing next to one of these the game can be saved until another day.

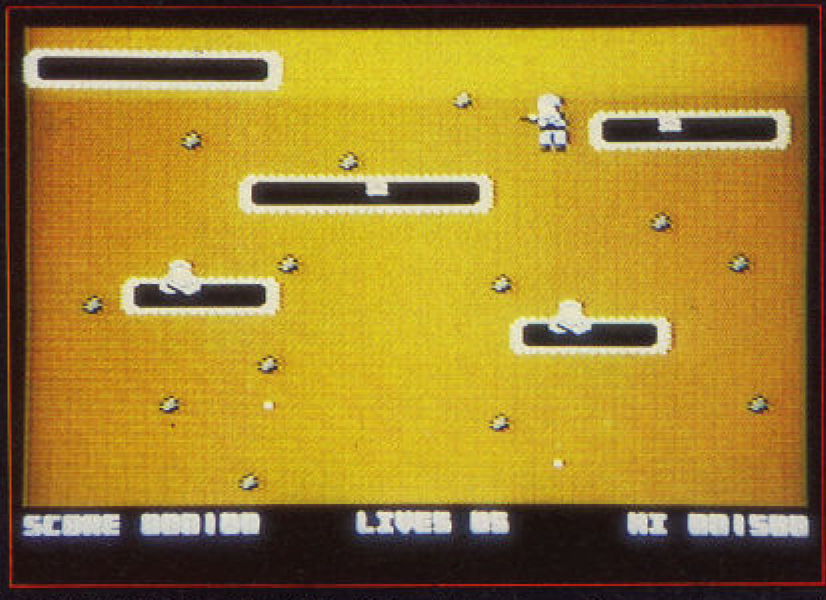
Graphically, the game is quite pleasant, the casual lope of the Doctor is particularly impressive. The challenges are difficult but logical and, apart from the nagging music, I wholeheartedly recommend this game.

J.G.

COMMANDO

Elite C-16, Plus/4 £7.95

Originality	Playability	Graphics	Value For Money
1	2	5	2



COMMANDO ON THE C64 was exstremely popular, it was full of scrolling screens and excellent sound effects. It would be very easy to expect that a conversion of the game by the same company would be of the same high quality. Well it isn't. The only similarity between this and the C64 version is the fact that you control a Commando who is under heavy enemy fire.

In this version of the game the screens are static. Before you can get on to the next level you have to clear the screen of enemy personnel.

The first level places your commando under heavy enemy fire from enemy trenches. I said heavy and I mean it, if you so much as pause for a second you will probably lose a life. There is a simple trick to this screen that enables you to get through it quickly. Dodge all of the enemy bullets and get behind the trenches in which the men hide.

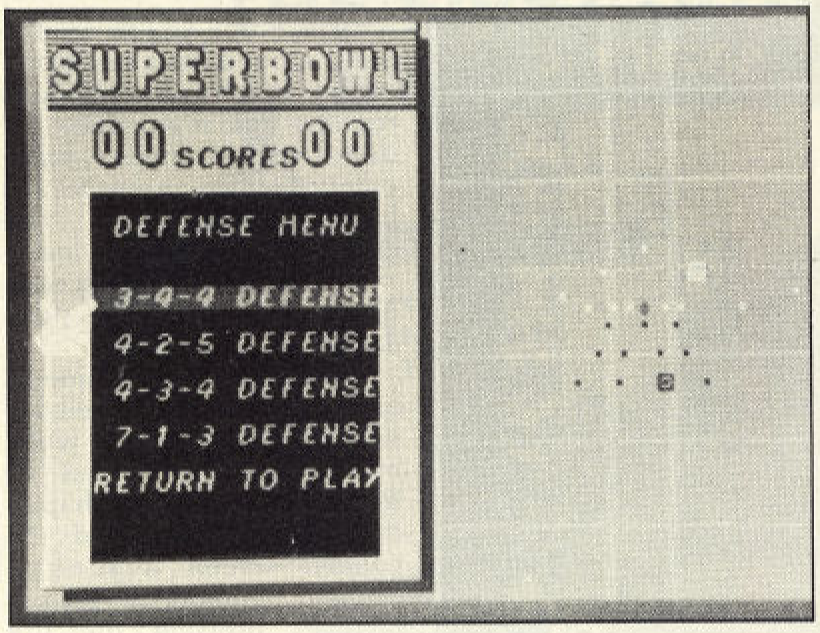
In the second screen our commando finds himself in front of a bridge guarded by the enemy. This is about as much as I can tell you. As soon as this screen opens, missiles and bullets are winging their way towards you, you don't stand a chance.

In my mind the only commando that should tackle this game is one with a death wish.

S.C.

SUPERBOWL
Ocean
C64

Originality 10 Playability 10 Graphics 10 Value For Money 10



THIRTY SIX. TWENTY FOUR. Thirty six. Hup! Hup! Yes, after considerable delays, Ocean's Superbowl has finally arrived amidst considerable razzamatazz and waving of pompoms. Most of the delays were due to various gameplay problems (scores of over 100 points were being recorded) but these have now been sorted out.

Fans of American Football will know that despite initial appearances, there is actually a great deal of skill involved in the game, organising tactics and running to predetermined patterns. Ocean has managed to pack a lot of detail into this game although some patriotic prejudice is displayed with a reference to real football in the instructions! For people who know nothing at all about this thrilling game, there is an audio tape included in the package which gives you a quick run down on the rules, terms and basic tactics. This works very well and is a lot better than the normal 12 page booklet that you usually get.

The screen is divided into two main sections. On the right is an overhead view of the playing area. This depicts the starting positions of the move that you want to try and execute and then follows the move through once the ball has been snapped. You control one previously determined player although it is possible to change this during the course of the play if you are fast enough. Eleven men on each side running in 22 different directions takes quite a bit of getting used to!

The left hand side of the screen serves two functions. As a purely decorative part of the game, an action replay of the previous play appears on the giant screen. This shows the players running, throwing, tackling and catching the ball and is nicely animated although it tends to get a bit repetitive after a while (you can stop the display with a quick press of the fire button).

The nitty gritty part of the game – deciding on your tactics is determined through a series of menus. Starting with the offense, an initial menu gives you the chance to try a long or short pass, a rushing play or a special play (field goal attempts, punts and goal line rushes). These choices lead into sub-menus giving you a choice of starting formations with such esoteric names as shotgun and split red. Selecting "view next frame" from the menu allows you to watch the players moving to their designated positions – a very useful option that gives you some understanding of the theory behind the manoeuvres. You can also change which player you want to receive the ball once lthe move has started.

When you are happy with your choices, the "play games" option allows your opponent (or the computer) to decide on their defensive strategy. As soon as that is ready, the move is ready to be executed. The center snaps the ball back to the quarterback whilst the other players start to move upfield according to your instructions. A quick press of the fire button moves the cursor around the eligible receivers starting with the one that you previously designated. As your finger is removed from the button, so the ball is thrown and the receiver comes under joystick control. He must then move to where he thinks the ball is going to land as he endeavours to catch it.

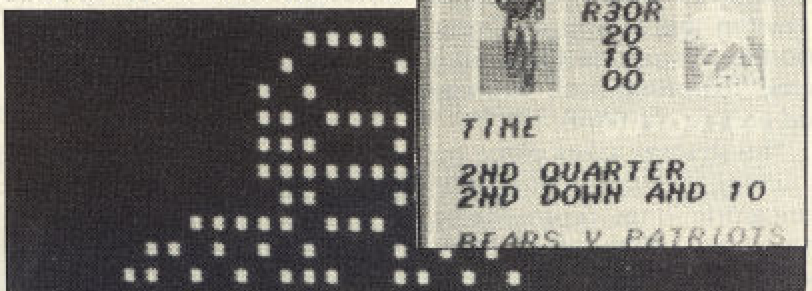
Instead of passing the ball, the offense may try to kick it, either attempting to score a field goal or simply punting to relieve their lines. Keeping the fire button pressed brings up a power scale showing the percentage of the maximum kick currently chosen. It is tempting to go for full range every time but this is wrong as the accuracy of a kick decreases with power.

The defense is somewhat more complicated to organise. Apart from choosing your initial formation from 3-4-4, 4-3-4, 4-2-5 and 7-1-3, you must also decide who is going to mark whom, which players are going to go for the opponent holding the ball and which offensive players will remain unmarked. Again, you can decide which player you wish to control in the ensuing move (number 73 William 'The Fridge' Perry is likely to be a popular choice).

Superbowl is the best American Football game seen to date. Based on this January's game in which the Chicago Bears thrashed the New England Patriots 46-10, it is an extremely credible situation. Devotees of the game need look no further. For people who know absolutely nothing about the game, why not try your hand before the new season starts on Channel 4.

G.R.H.

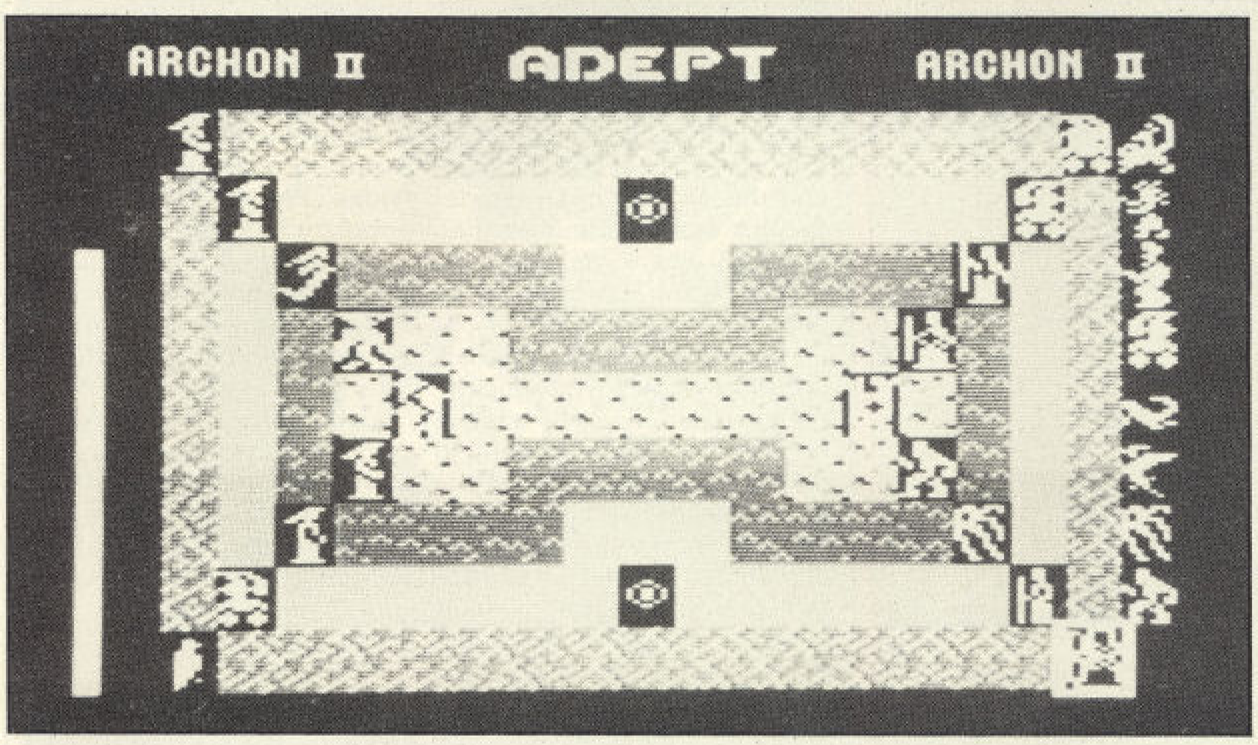




ARCHON II: ADEPT

Ariolasoft
£12.95, joystick required, C64

Originality 10 Playability 10 Graphics 10 Value For Money 10



ARCHON WAS ONE OF THE most original strategy games ever written, justifiably winning several awards. Now Electronic Arts has released a sequel on the Ariolasoft label. Archon II, Adept sees you taking sides in an epic struggle between the Master of Order and the Mistress of Chaos in a game featuring both strategy and arcade elements.

The strategy takes place on a screen featuring the four concentric rings representing the classical elements of Earth Water, Air and Fire. In addition, there are two neutral squares which represent the void and the home squares for each side – the Fortress of Order and the Temple of Chaos.

The aim of the game is to occupy six power points. Two of these are the void squares and the other four are the outer corners of the elemental bands. These four rotate in turn from band to band. You can also win by the total annihilation of the opposition's forces.

You start the game with four adepts – one in each element. Each turn, they can either move or cast a spell providing that you have sufficient energy to carry out your choice. There are seven spells to choose from but the one that you will use more than any other is "summon". This is used to bring another piece on the board.

The other spells available to you are; heal one of your pieces, weaken an opponent's piece, imprison an enemy, release one of your own imprisoned pieces, banish a hostile enemy or something called apocalypse which is a final battle used to put your adversary out of his misery. Casting spells costs varying amounts of energy depending on its potency. How much energy you have at your disposal depends on how many power points you occupy.

There are two types of pieces that can be summoned, demons and elementals. Both sides have the same demons at their disposal – juggernauts, wraiths, gorgons and chimera whilst their elementals are different. Order can call on the services of a giant, kraken, thunderbird and salamander, representing earth, water, air and fire whilst Chaos has a behemoth, siren, ifrit and firebird available to him.

All these characters have different strengths and weaknesses when it comes to combat. Sirens for example just have to sing and the opposition starts to die. Salamanders hurl fireballs, gorgons paralyse whilst wraiths get stronger as you get weaker – they are also invisible most of the time and so make extremely formidable opponents. Not quite as bad as a juggernaut though which is best described as pure energy on wheels. It just steamrollers opposition out of the way.

Combat occurs when two pieces want to occupy the same square. The scene switches to the battleground where you must make instant decisions as you ltry to probe the opposition's weaknesses and utilise your own strengths to their best advantage. Each piece's strength is displayed as an energy bar down the side of the screen. This reduces for each successful wound inflicted. When the bar reaches zero, the icon dies leaving the victor in sole possession of the disputed square.

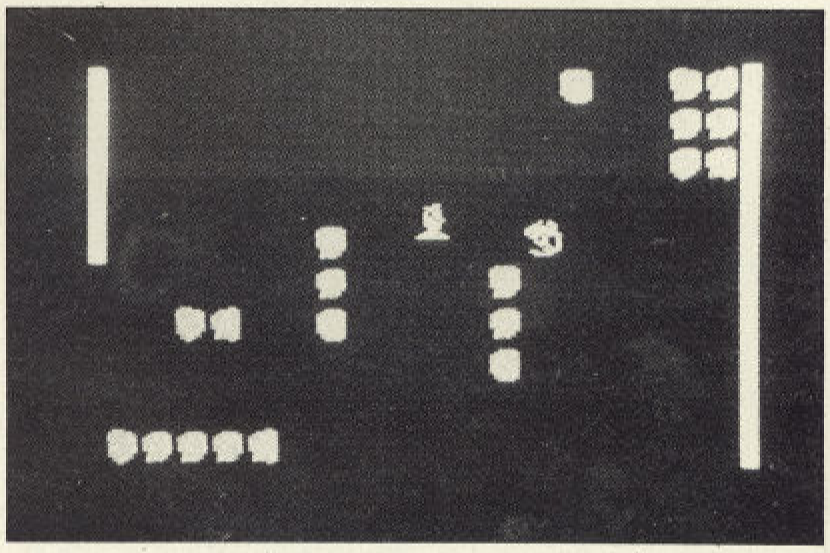
As might be expected, pieces fight best when in their home element e.g. krakens in the water band. After you have fired your thunderbolt or whatever, it takes time before you are allowed to fire your next. This time interval varies from piece to piece and the computer lets you know with a ping – high or low depending on which side you're on. The battle ground has a number of barriers which must be dodged round or used strategically. The different elements have differing effects on missiles and icons. For example, fire wounds an icon but leaves missiles untouched whilst earth destroys missiles and slows icons.

Control of the game is entirely via a joystick. Spells are selected from a menu whilst movement is achieved by moving a square shaped cursor. Moving round the combat screen is straightforward. Aiming a missile involves pressing the fire button and moving the joystick in the desired direction. Adepts can move their missiles whilst in flight – a useful trick to know.

Archon II features a wide range of options to choose from: Which side you play, number of players and their skill levels. Be warned though, the computer plays a very mean game and you are likely to be thrashed in your first few games. I would strongly recommend that you watch the demonstration games a few times so that you can get some idea of the strategies and tactics required.

Archon II is an excellent strategy game and one that will take you a lifetime to master – then you can play with the other side and learn a totally new set of tactics. Whilst it doesn't quite reach the exalted standards set by the original, that is no real criticism and the game can be unreservedly recommended.

G.R.H.



Save your fingers a lot of work with our new software service.

SOFTWARE FOR SALE

IT'S THREE O'CLOCK IN THE MORNING. You sit at the computer keyboard having just finished a marathon typing session entering one of the superb programs from Your Commodore. Your fingers reach for the keyboard and press the letters R, U and N/. You sit back expectantly and...nothing happens.

Well, I'm sure that we have all had problem before now. When it does happen it's a matter of spending hours searching through the program for any typing mistakes. No matter how long you look or how many people help you, you can usually guarantee that at least one little bug slips through unnoticed.

Here, at Your Commodore, we pride ourselves on the quality of listing that we print. Unfortunately, this usually means that they are also very long, thus taking longer to type in and leaving more room for errors. All of the listings in Your Commodore are taken straight from a printout of working programs, it is therefore very unusual for errors to appear in the magazine.

Because of the length of our programs we do get a large number of requests from readers who would like us to put specific programs on tape or disk for them. Obviously this is very time consuming and means that we can't spend as much time working on the magazine as we would like.

We are therefore proud to announce the start of the 'Your Commodore Software Service'. Most of the programs from each issue of the magazine will now be available on a single cassette for a price of just £4.00. We will not be making disks available since they would have to be a lot more expensive and more difficult to post. This shouldn't cause you any problems though as none of the programs will be protected and it will be a simple matter to save the programs to disk yourselves.

All programs on the cassette will be saved using a tape turbo routine. However, we cannot guarantee that all programs will work correctly with this turbo routine present. We therefore recommend that before you use any of the programs you make a copy of the programs on your own cassette or disk and use this version of the program **not** the original.

PLEASE COMPLETE IN BLOCK CAPITALS

ORDER FORM

All cheques/postal orders should be made payable to:-
ARGUS SPECIALIST PUBLICATIONS LTD

I would like to order the following cassettes: (please tick box)

UTILITY SPECIAL	YCMA862	£4.00	
APRIL '86	YCAPR86	£4.00	
MAY '86	YCMAY86	£4.00	
JUNE'86	YCJUN86	£4.00	

I enclose a cheque/postal order for £ ....

NAME..

ADDRESS..

....

....

POSTCODE..

All orders should be sent to:-
YOUR COMMODORE SOFTWARE, READERS SERVICES, ARGUS SPECIALIST PUBLICATIONS, WOLSEY HOUSE, WOLSEY ROAD, HEMEL HEMPSTEAD, HERTFORDSHIRE HP2 4SS.

Please allow at least 28 days for delivery.

SPRITE IDEAS

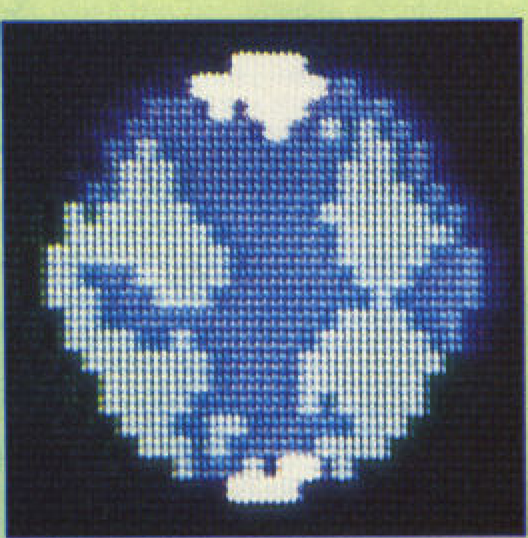
When you are designing a game one of the longest jobs is designing the sprites. If you are good at art then fine, if not your next monster will probably end up looking like a square box with legs.

Now, Your Commodore comes to the rescue once again with Sprite Ideas. If you have designed any sprites for games and you don't mind other people seeing your masterworks then why not send them into us. Each month we will be offering £10 for the best entries.

Your sprites can be anything at all (within reason), if you've designed a series of animated characters then send in the lot. We'd love to have a look at them.

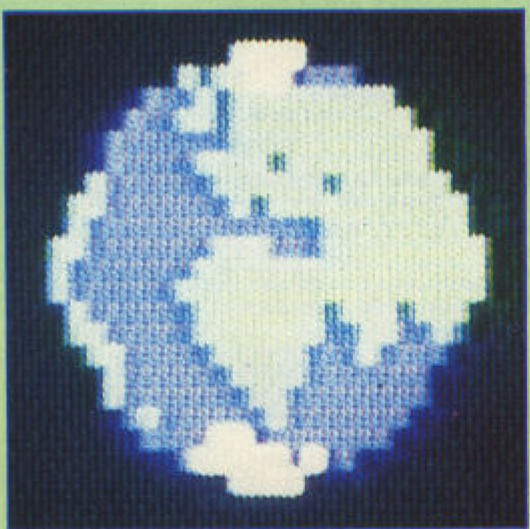
So, next time you are after an Ogre to put in your new game, have a look in this section of the magazine and you may find just what you are looking for.

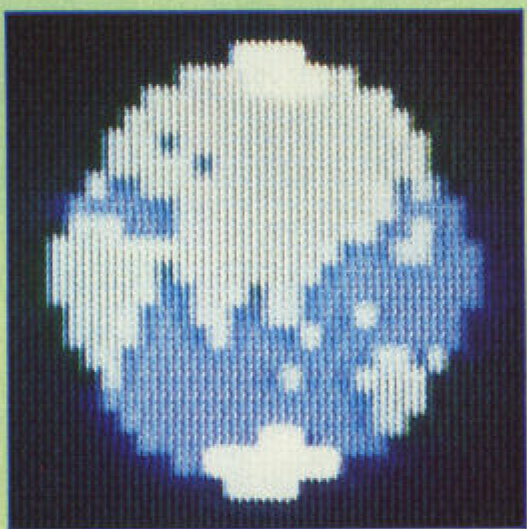
This month's sprites are from Aman Khan from Hayes, Middlesex.











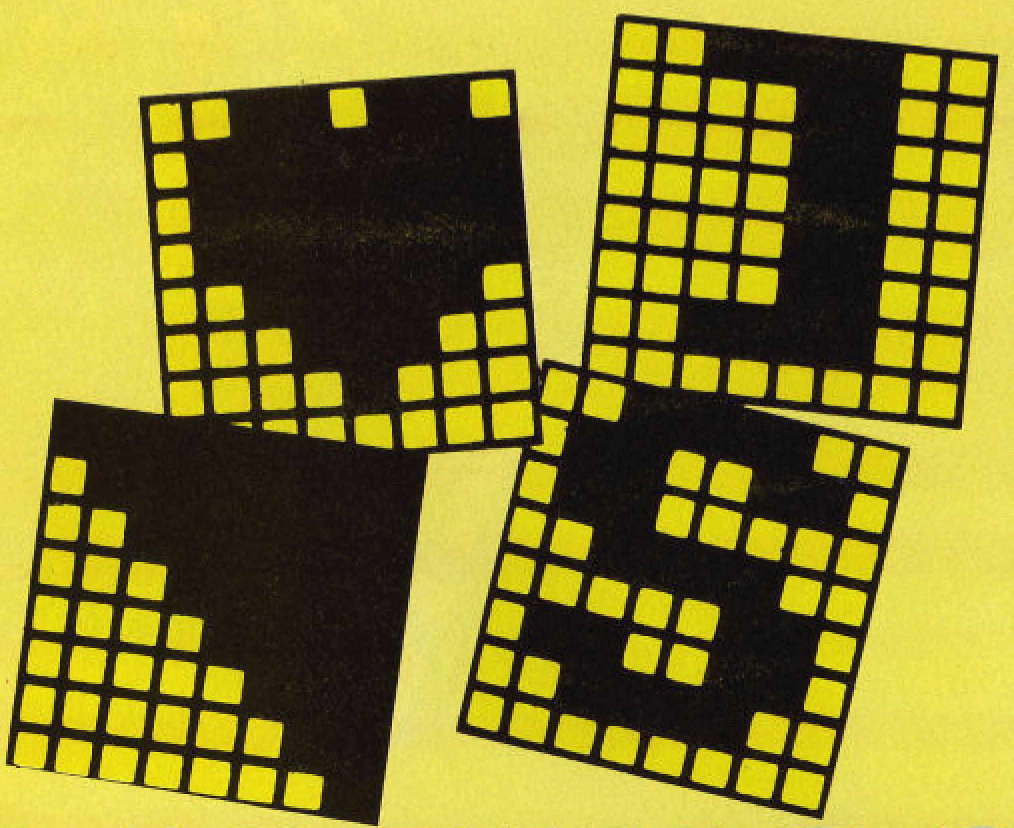




```
5 POKE53281,0:POKE53280,0:PRINT"(CLR)":REM SETS UP SCREEN COLOURS AND CLEARS SCR
EEN
10 FORA=0TO1407:READQ:POKE12288+A,Q:NEXT:REM POKES DATA INTO 12288 ONWARDS.
20 REM ONCE DATA HAS BEEN READ IN ONCE TYPE 'RUN 1200' TO RUN THE PROGRAM AGAIN
100 REM FRAME ONE...SPRITE ONE
105 DATA0,0,10,0,3,105,0,63,213,0,253,85,3,255,117,3,255,245,15,255,245,23,255
110 DATA213,21,255,85,85,117,85,85,253,85,87,255,85,31,255,85,23,255,213,5,255
120 DATA85,1,127,85,1,125,85,0,93,214,0,21,90,0,1,106,0,0,10,0
130 REM FRAME ONE...SPRITE TWO
135 DATA160,0,0,213,192,0,93,252,0,125,127,0,85,255,192,87,255,192,87,255,240
140 DATA85,215,252,85,85,124,87,245,127,95,253,95,95,255,255,87,255,252,85,255
150 DATA220,85,127,80,85,127,64,85,93,64,85,85,0,153,84,0,170,64,0,160,0,0,0
200 REM FRAME TWO...SPRITE ONE
205 DATA0,0,10,0,1,93,0,21,85,0,245,87,3,213,85,3,221,85,15,253,85,63,245,85
```

```
210 DATA63,245,85,255,85,85,117,85,85,253,85,87,63,85,85,63,213,85,15,213,85
220 DATA3,85,85,3,117,85,0,85,85,0,21,106,0,1,170,0,0,10,0
230 REM FRAME TWO...SPRITE TWO
235 DATA160,0,0,107,192,0,223,252,0,223,255,0,95,255,192,255,255,192,127,255
240 DATA240,93,127,252,85,87,252,127,87,255,255,255,255,255,253,247,255,253,244
250 DATA127,245,212,127,245,80,95,245,64,87,213,64,85,213,0,89,84,0,170,64,0
260 DATA192,0,0,0
300 REM FRAME THREE...SPRITE ONE
305 DATA0,0,10,0,3,126,0,21,223,0,87,223,1,85,95,1,85,255,13,85,127,61,85,93
310 DATA53,85,85,245,85,127,213,85,255,213,87,255,53,85,255,61,85,255,15,85,127
320 DATA3,85,95,3,85,87,0,117,85,0,21,105,0,1,170,0,0,10,0
330 REM FRAME THREE...SPRITE TWO
335 DATA160,0,0,149,64,0,255,252,0,255,255,0,255,255,192,127,255,192,253,255
340 DATA240,255,255,252,87,255,252,87,255,255,255,255,255,255,255,255,253,253
350 DATA252,253,125,116,245,117,80,245,85,64,213,85,64,213,85,0,85,84,0,153,64
360 DATA0,160,0,0,0
400 REM FRAME FOUR...SPRITE ONE
405 DATA0,0,10,0,3,235,0,55,255,0,247,255,1,87,255,1,255,223,5,127,253,21,93
410 DATA127,21,85,87,85,127,87,85,255,255,87,255,247,21,255,245,21,255,213,5
420 DATA127,213,1,127,85,1,93,85,0,93,85,0,21,90,0,2,170,0,0,10,0
430 REM FRAME FOUR...SPRITE TWO
435 DATA160,0,0,235,192,0,255,252,0,255,255,0,255,255,192,255,255,192,255,255
440 DATA240,255,255,244,255,255,244,255,255,85,255,253,85,255,245,85,253,245,84
450 DATA125,93,84,117,85,240,85,117,192,85,87,192,85,87,0,85,84,0,169,64,0,160
460 DATA0,0,0
500 REM FRAME FIVE...SPRITE ONE
505 DATA0,0,10,0,3,254,0,63,255,0,255,255,3,247,255,3,255,127,13,127,255,21,87
510 DATA255,63,87,255,255,255,255,255,247,255,255,245,253,63,213,245,63,213,117
520 DATA15,85,85,3,85,85,1,85,85,0,85,85,0,21,85,0,1,106,0,0,10,0
530 REM FRAME FIVE...SPRITE TWO
535 DATA160,0,0,171,192,0,255,252,0,255,255,0,255,255,192,255,255,192,255,253
540 DATA208,255,253,84,255,245,212,255,87,213,253,85,85,245,85,85,245,117,84,93
550 DATA85,84,85,95,112,117,127,192,85,95,192,85,95,0,165,84,0,170,64,0,160,0,0
560 DATA0
600 REM FRAME SIX...SPRITE ONE
605 DATA0,0,10,0,3,235,0,63,255,0,255,255,3,255,255,3,127,255,5,255,255,21,255
610 DATA255,63,255,255,255,255,255,255,127,255,255,95,125,61,93,125,53,85,87,5
620 DATA85,85,1,85,93,1,85,85,0,85,85,0,21,90,0,1,170,0,0,10,0
630 REM FRAME SIX...SPRITE TWO
635 DATA160,0,0,175,192,0,255,244,0,255,245,0,255,253,64,255,255,192,255,247,80
640 DATA255,85,87,253,117,87,213,245,85,85,85,85,85,85,85,117,85,84,85,85,84,95
650 DATA93,80,127,255,64,127,255,64,95,221,0,85,84,0,170,64,0,160,0,0,0
700 REM FRAME SEVEN...SPRITE ONE
705 DATA0,0,10,0,3,254,0,63,255,0,255,255,3,255,255,3,255,255,15,255,255,63,255
710 DATA255,63,255,253,255,255,213,127,255,85,95,125,85,29,125,117,21,87,85,5
720 DATA85,95,1,93,127,1,85,127,0,85,95,0,21,85,0,1,106,0,0,10,0
730 REM FRAME SEVEN...SPRITE TWO
735 DATA160,0,0,233,64,0,253,84,0,253,87,0,255,87,192,255,223,192,247,87,240,85
740 DATA85,124,117,85,124,245,85,93,85,85,95,85,85,127,85,85,124,85,85,92,93,85
750 DATA80,255,85,64,255,93,64,221,117,0,85,84,0,169,64,0,160,0,0,0
800 REM FRAME EIGHT...SPRITE ONE
805 DATA0,0,10,0,3,234,0,63,253,0,255,253,3,255,255,3,255,255,15,255,247,63,255
810 DATA85,63,253,117,255,213,245,255,85,85,253,85,85,61,117,85,23,87,117,5,127
820 DATA253,1,127,253,1,95,117,0,85,85,0,21,86,0,2,170,0,0,10,0
830 REM FRAME EIGHT...SPRITE TWO
835 DATA160,0,0,170,64,0,85,84,0,87,247,0,95,255,192,215,255,192,85,127,240,85
840 DATA127,252,85,127,252,85,95,245,85,87,85,85,87,245,85,95,252,85,95,252,85
850 DATA87,240,117,87,192,213,85,192,85,85,0,149,84,0,169,64,0,160,0,0,0
900 REM FRAME NINE...SPRITE ONE
905 DATA0,0,10,0,1,106,0,61,85,0,253,87,3,255,95,3,255,215,15,247,85,63,85,85
910 DATA61,85,85,213,85,85,85,85,85,85,85,85,23,117,85,63,253,85,15,253,117,3
920 DATA117,213,1,85,85,0,85,85,0,21,105,0,1,170,0,0,10,0
930 REM FRAME NINE...SPRITE TWO
935 DATA160,0,0,85,64,0,85,124,0,247,253,0,255,255,64,255,255,192,127,255,240
940 DATA127,255,212,127,255,84,95,245,85,87,85,85,87,253,85,127,255,84,95,255
950 DATA84,95,253,80,87,245,64,87,213,64,85,221,0,101,84,0,169,64,0,160,0,0,0
1000 REM FRAME TEN...SPRITE ONE
1005 DATA0,0,10,0,1,85,0,21,85,0,87,247,3,95,255,3,215,255,7,85,127,21,85,127
1010 DATA53,85,127,245,85,95,85,85,87,85,85,87,53,85,127,61,85,95,13,117,95,1
1020 DATA213,87,1,85,87,0,85,85,0,21,89,0,1,170,0,0,10,0
1030 REM FRAME TEN...SPRITE TWO
1035 DATA160,0,0,149,64,0,127,212,0,253,85,0,255,117,64,255,245,64,255,245,80
1040 DATA255,213,92,255,85,84,245,85,85,85,85,85,253,85,95,255,85,124,255,85
1050 DATA124,253,85,80,245,85,64,213,85,64,221,85,0,85,84,0,105,64,0,160,0,0,0
1100 REM FRAME ELEVEN...SPRITE ONE
1105 DATA0,0,10,0,1,85,0,21,127,0,247,253,3,255,255,3,255,255,5,127,255,21,127
1110 DATA255,21,127,255,85,95,245,85,87,85,85,95,245,21,127,255,21,95,255,5,95
1120 DATA253,1,87,245,1,87,213,0,85,221,0,21,85,0,1,106,0,0,10,0
1130 REM FRAME ELEVEN...SPRITE TWO
1135 DATA160,0,0,105,64,0,213,220,0,85,125,0,117,85,192,245,87,192,245,87,240
1140 DATA213,85,212,85,85,84,85,87,245,85,95,253,85,95,255,85,87,252,85,85,252
1150 DATA85,85,112,85,85,64,85,85,64,85,85,0,165,84,0,170,64,0,160,0,0,0,0
1200 REM THIS SETS UP THE PARAMETERS.
1205 V=53248:POKEV+21,3:POKEV+39,1:POKEV+40,1:PRINT"(CLR)(YEL)":POKE53281,0:POKE
53280,7
1210 POKEV+23,3:POKEV+28,3:POKEV+37,6:POKEV+38,5
1220 POKEV,170:POKEV+2,194:POKEV+1,125:POKEV+3,125
1230 A=192
1235 REM THIS PART SPINS THE PLANET.
1240 POKE2040,A:POKE2041,A+1
1250 FORB=0TO125:NEXT:A=A+2:IFA=214THEN1230
1260 GOTO1240
```

Listings will be much easier to enter with our new system.



LISTINGS

COMMODORE LISTINGS ARE RATHER well known for the horrible little black blobs that always abound. Unfortunately the graphics characters which are used to represent graphic and control characters do not reproduce very well and they are also difficult to find on the Commodore keyboard.

In future all control and graphics commands will be replaced by a mnemonic within square brackets. This mnemonic is not typed out as printed in the magazine but rather the corresponding key or keys on the keyboard are pressed. For example [RIGHT] means press the cursor right key, you do not type in [RIGHT]. All of the keywords, what keys to press and how they are shown on the screen are shown below.

Any character that is accessed by pressing shift and a letter will be printed as [Sletter].

[SA] shift and A

[S+] shift and +

Any character that is accessed by pressing the Commodore key and a letter will be printed as [Cletter]

[CA] Commodore and A

[C+] Commodore and +

[C1] Commodore and 1

If any characters are repeated the mnemonic will be followed by a number. This number is how many times you should enter the character. Any number of spaces over one will also be represented in this form

[RIGHT10] press cursor right 10 times

[C+10] press Commodore and + 10 times

[SPC10] Press the space bar 10 times

Any other characters should be easily recognisable for example CTRL-N means press CTRL and N and LEFT-ARROW means press the left arrow.

Any number of mnemonics can be enclosed in brackets for example

[SA10,SPC10,SA10]

means type 10 shift A's 10 spaces and another 10 shift A's.

Mnemonic	Symbol	what to press
[RIGHT]		left/right
[LEFT]		shift left/right
[UP]		Shift & up /down
[DOWN]		up/down
[F1]		f1
[F2]		shift & f1
[F3]		f3
[F4]		shift & f3

Mnemonic	Symbol	what to press
[F5]		f5
[F6]		shift & f5
[F7]		f7
[F8]		shift & f7
[CLEAR]		shift & CLR /HOME
[HOME]		CLR/HOME
[RVSON]		CTRL & 9
[RVSOFF]		CTRL & 0

Mnemonic	Symbol	what to press
[BLACK]		CTRL & 1
[WHITE]		CTRL & 2
[RED]		CTRL & 3
[CYAN]		CTRL & 4
[PURPLE]		CTRL & 5
[GREEN]		CTRL & 6
[BLUE]		CTRL & 7
[YELLOW]		CTRL & 8









THE CREATOR SERIES

Now you can let your mind go in a crescendo of creativity. You may think you are unable to program, you may know nothing about computing but now YOU are able to write stunning original machine code programs. ARCADE CREATOR, the first release in the CREATOR SERIES, allows you to program your own arcade games in the language you know best – English.

If you prefer the complex mind stretching world of the strategy program, then the second in the series, BATTLE CREATOR, is for you. Using simple commands you can devise your own brain buster, without busting your own brain in the process.

For the budding musician, the third in the series, MUSIC CREATOR, helps you to compose your own melodies and rhythms.

The Argus Press Software Group Ltd
Liberty House, 222 Regent Street, London W1R 7DB
Telephone 01-439 0666

Music Creator – CBM64

THE CREATOR SERIES

UNLEASHES THE POWER OF YOUR IMAGINATION

Arcade/Battle Creator –
CBM64, Spectrum & Amstrad

AN ARGUS SPECIALIST PUBLICATION

April 1986

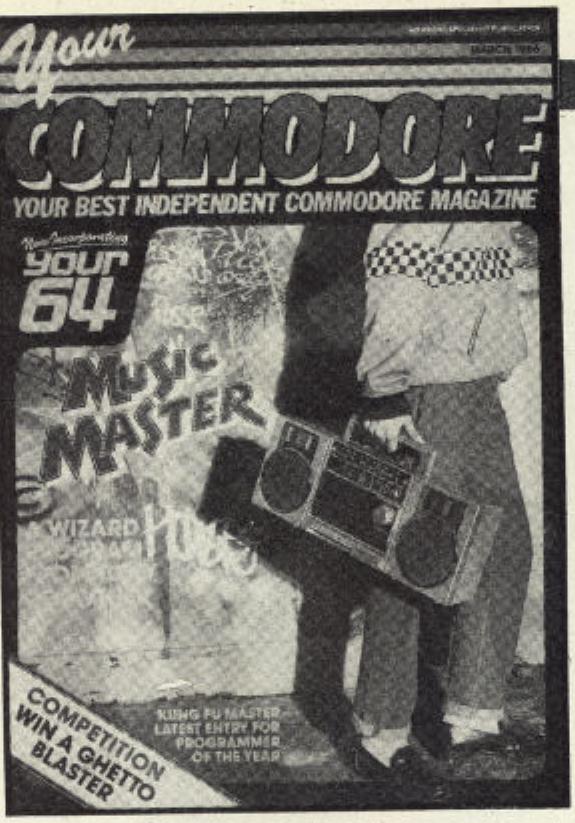
£1.00

Your COMMODORE READER'S SURVEY









1. Would you please tick the box against the statement which best describes how much of Your Commodore you normally read or look through:
Read or look through most or nearly all the articles/features ☐
Read or look through some of the articles/features ☐
Just read or look through the occasional article/feature ☐

2. With regard to the advertisements in Your Commodore, do you:
Read or look through most or nearly all of the advertisements? ☐
Read or look through some of the advertisements? ☐
Just read or look through the occasional advertisement? ☐
Very rarely/never look at the advertisments? ☐

3. Thinking specifically about the advertising content of Your Commodore, would you please rate the two main types of advertising matter – Display and Classified – in terms of usefulness (please tick one against each type):

	Display	Classified
Very useful	☐	☐
Useful	☐	☐
Not very useful	☐	☐
Not at all useful	☐	☐

4. Have you ever ordered or bought equipment/products after reading an advertisement in Your Commodore?
Regularly ☐
Occasionally ☐
Never ☐
If the answer to Question 4 is yes, what was the last item you purchased in this way and what was its value? ....

5. Does anyone else read your copy of Your Commodore?
No ☐
1 or 2 ☐
3 or 4 ☐
More than 4 ☐

6. Do you keep your copies of Your Commodore for:
One month? ☐
Three months? ☐
Six months? ☐
A year or more? ☐
IF KEPT, PLEASE ANSWER THE NEXT QUESTION.

7. How often do you refer to back issues of Your Commodore?
Once a week or more often ☐
About once a month ☐
Once every three months ☐
Less often ☐
Never refer to back issues ☐

8. What magazines other than Your Commodore's competitors do you read?

....
....
....
....
....
....

9. What Daily newspaper do you regularly read?
Daily Mail ☐
Daily Express ☐
Daily Mirror ☐
The Sun ☐
Today ☐
The Guardian ☐
The Times ☐
The Daily Telegraph ☐
Financial Times ☐

10. What Sunday newspaper do you regularly read?
Sunday Times ☐
Sunday Telegraph ☐
The Observer ☐
Sunday Express ☐
Mail on Sunday ☐
News of the World ☐
Sunday People ☐

11. Name the three television programmes you view most regularly.

....
....
....

12. Which computer(s) do you own?
C16 ☐
Plus/4 ☐
C64 ☐
C128 ☐
Vic 20 ☐
PET ☐
Spectrum ☐
Amstrad ☐
BBC ☐
Electron ☐
Atari ☐

13. Do you own one of the following disk drives?
1541 ☐
1551 ☐
1570 ☐
1571 ☐

14. Do you own any of the following printers?
Commodore printer ☐
Epson-compatible printer ☐
Other ☐

15. Do you own any of the following peripherals?
Joystick ☐
Lightpen ☐
Mouse ☐
Graphics pad ☐

16. How long have you had a Commodore computer?
Less than three months ☐
Three to six months ☐
Seven months to one year ☐
One year to two years ☐
Over two years ☐

17. Do you use your computer for the following:

Original programmings	All the time	More than half the time	Sometimes	Never
Typing in games listings				
Typing in utility listings				
Playing games				
Educational uses				
Business uses				

18. Who else uses your computer?
Nobody ☐
Spouse ☐
Parent ☐
Children ☐
Friends ☐
Other ☐

19. How much do you estimate you have spent in total in the last 12 months on your computing activities?
0-£50 ☐
£51-100 ☐
£101-200 ☐
£201-500 ☐
£500-£1000 ☐
£1000-1500 ☐
£1500-2000 ☐
Over £2000 ☐

20. How much do you expect to spend on hardware over the next year?
0-£50 ☐
£51-£100 ☐
£101-200 ☐
£201-£500 ☐
Over £500 ☐

21. How much do you normally spend in a 12-month period on the following types of software?

22. Do software reviews influence your buying?
Yes ☐
No ☐

	Games	Business Software	Educational Software	Utilities
0-£50				
£51-£100				
£101-£200				
£201-£300				
£301-£500				

23. Do Software Charts influence your buying?
Yes ☐
No ☐

24. Please tick the box which best describes you:
New reader (within the last 3 months) ☐
Established reader ☐
Occasional reader ☐

25. Were you previously a regular reader of Your 64?
Yes ☐
No ☐

26. Were you previously a regular reader of Your Commodore before we incorporated Your 64? Yes ☐
No ☐

27. Were you previously a regular reader of BOTH Your Commodore and Your 64?
Yes ☐
No ☐

28. Since we incorporated Your 64, do you think that Your Commodore is:
Better ☐
Same ☐
Worse
PLEASE STATE WHY ....
....

29. What do you think about the balance of articles in Your Commodore?

	More	About right	Less
New	☐	☐	☐
Programming articles	☐	☐	☐
Software reviews	☐	☐	☐
Hardware reviews	☐	☐	☐
Book reviews	☐	☐	☐
Games to type in	☐	☐	☐
Utilities to type in	☐	☐	☐
Business page	☐	☐	☐
Adventure column	☐	☐	☐
Letters	☐	☐	☐
Competitions	☐	☐	☐

What else would you like to see in Your Commodore? ....
....

30. Which listings do you type in?

	All	Some	None
Games	☐	☐	☐
Utilities	☐	☐	☐

31. Which other computer magazines do you regularly buy?
Commodore Horizons ☐
Commodore User ☐
Commodore Computing International ☐
Zzapp 64 ☐
Compute ☐
Your Computer ☐
Popular Computing Weekly ☐
Personal Computer World ☐
Other ☐

32. Are you aware of Your Commodore's scheduled publication date?
Yes ☐
No ☐

33. If you answered yes to Question 32, do you attempt to purchase the magazine on that day?
Yes ☐
No ☐

34. Do you normally obtain your copy by:
Casual purchase
Newsagent home delivery ☐
Newsagent shop collection ☐
Subscription ☐

35. If you do not obtain your copy be subscription, is it due to one of the following?
Subscription too expensive ☐
Good availability through local newsagent ☐
Not every issue is required ☐
Have subscribed previously but lapsed ☐

36. If you do not subscribe, from which type of newsagent do you most often obtain your copy?
High Street shop ☐
Estate shop ☐
Travel Point ☐
Corner shop ☐

37. Are you a member of a computer club?
Yes ☐
No ☐
If yes, please give details ....
....
....
....

PERSONAL DETAILS

38. Please tick the box which represents the annual total of your NET income (i.e. after tax, National Insurance, pension contributions, etc):
From £11690+ ☐
From £9350 to £11680 ☐
From £7480 to £9340 ☐
From £5980 to £7470 ☐
From £4780 to £5970 ☐
Name ....
Address ....
....
....

Marital status ☐
Sex ☐
Age ☐
Occupation ☐
Number of children ☐

We would like to thank you very much for your help. The information will be regarded as confidential and will be used solely for the purpose of the survey.

TUCK INTO A
THIRD FOLD

Postage will be paid by licensee

Do not affix Postage Stamps if posted in Gt Britain, Channel Islands, N Ireland or the Isle of Man

BUSINESS REPLY SERVICE
Licence No. WC 3970

Your Commodore
Argus Specialist Publications Limited
No. 1, Golden Square,
LONDON W1R 3AB

2

FIRST FOLD

SECOND FOLD

£10,000 HOLIDAY BONANZA





SPRING SALE NEW LOW PRICES

Maintenance, Service, Support, Training all from your No 1 Commodore centre.

WELCOME TO THE WORLD OF COMMODORE

PRINTERS

DPS 1101 Daisywheel 295.00

MPS 1000 NLQ £238.00
DOT MATRIX

Comprint Interface 61.99

128D Business Package

containing:

128D + 80 Col Monitor + Printer
Including 5 Disks + Box Paper

Inc. Superscript.

£875 inc VAT

C128D with Monitor

Internal D/Sided
1571 Disk Drive
With 80 Column
Monochrome
Monitor

£565.00

MONITORS

1900M:80 COLM 145.00
1902 COL Monitor 285.00

PHILLIPS
Green/Amber.................95.00
Phillips Cable...............10.00
128 Cable 40/80 Colm........14.95

Yes Its true !!!

For a limited period we are giving away a £5.00 Travel Chek for spending money on your next holiday with EVERY £50 SPENT.
Thats not all folks !!!
your vouchers could also win you a FREE stay HOLIDAY for two or four people in our lucky numbers draw.

ONE ENTRY FOR EVERY £5 VOUCHER

64 CORNER

C64 Compendium 189.00
C2N Recorder 42.00

C128

The 3 in 1 computer
100% compatible with C64 software
128K RAM, 72K ROM, 6510 & Z80A CPU's

£265.00

SOFTWARE

Superscript 64 - Disk	49.00
Superscript 128 - Disk	74.00
Superscript + Superbase 64 - Disk	99.00
Superbase 64 - Disk	59.00
Superbase 128 - Disk	79.00
Superbase + Superscript 128 - Disk	139.00

C128 and 1570 Package

C128 computer
1570 disk drive
5 × diskettes

£445.00

MEDIA

Verbatim which are the best disks:

SS DD 40T Diskettes	17.95
DS DD 40T Diskette	24.75
SS DD 80T	24.75
DS DD 80T Diskettes	32.75

PAPER

11 × 9.5 (2000 sheets)	12.65
11 2/3 × 9 1/4	11.95

LABELS

3.5 × 1 7/16 (1000)	4.50
3.5 × 1 7/16	16.00

TO CLEAR!

MPS 801 (New) 129.00
MPS 801 (Shop Soiled) 99.00
C64 (Shop Soiled) 99.00

CHROMASONIC Computer Centres

48 Junction Road, Archway, London N19 5RD
01-263-9493/5
238 Muswell Hill Broadway, London N10 3SH
01-883-3705

128 DISC DRIVES

1570 single sided disk drive 195.00
1571 double sided disk drive 265.00

£189.00



MODEM with 3 month subscription 75.00
MIRACLE MODEM 113.25
HI-RES GRAPHICS PAD 56.00

GRAND 128 TRADE IN/PART EXCHANGE

* CONDITIONS OF REDEMPTION OF TRAVELLERS CHEKS PRINTED ON EACH VOUCHER

Access "INSTANT CREDIT" "INSTANT CREDIT" VISA

Terms

All products are guaranteed for one year unless otherwise stated. Payment may be made by Access, Barclaycard, Bankers draft, Building Society cheque, cash or postal order. Sorry, cheques need five days for clearance. We reserve the right to change prices without prior notice. **All prices are inclusive of VAT.** Please check before ordering for carriage charges.



GAME of the month

Gordon Hamlett explores the complexities of US Gold's Ultima IV.

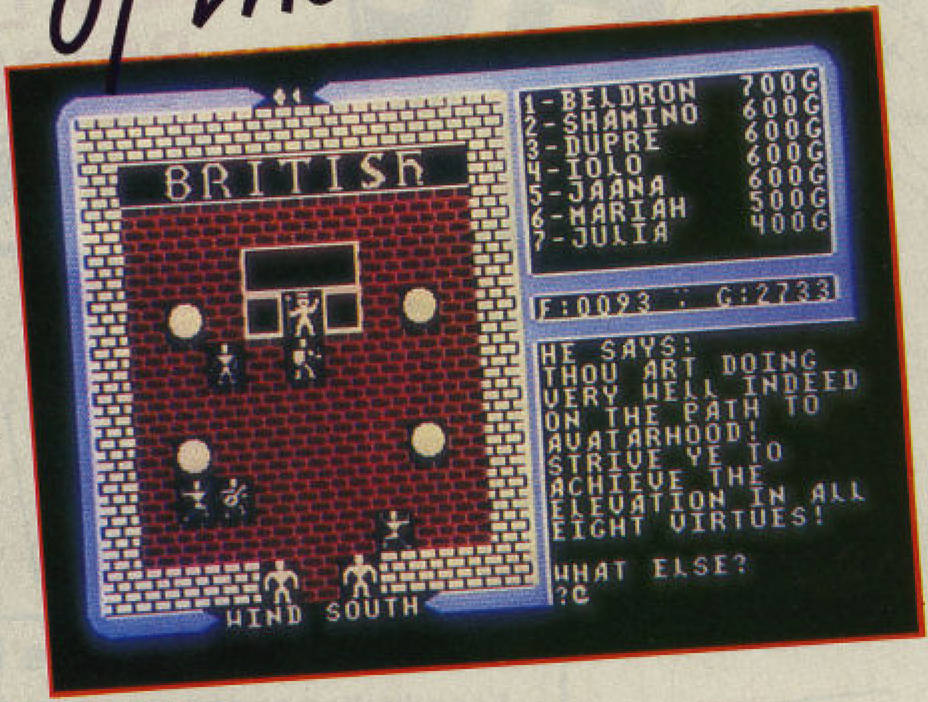
SINCE THE MENACE OF EXODUS WAS destroyed, Lord British is looking for someone to improve the quality of life throughout the land of Britannia. This involves finding or achieving something called Avatar about which, everybody has heard and nobody appears to know anything.

How you came to be volunteered for this quest is an interesting story. Whilst walking in the countryside, you see a bright light and hear a strange noise. From then on, things get curiouser and curiouser and you are inexorably lured by some music into a gipsy caravan. There, an old woman asks you some questions based on honour, valour, justice and other virtues. Depending on your answers, your character is assessed and your profession determined. This may be one of eight types including fighter, mage, druid and ranger.

The land of Britannia is a large and varied place. There are eight major towns, each one specialising in one of the eighth professions. The people inside are friendly and you will need to talk to everyone you meet, trying to glean useful scraps of information. There are castles and villages to be explored too – if you can find them. Outside of the towns, the land is more hostile – wandering bands of orcs, rogues, trolls and two-headed ettins etc. roam the countryside intent on trying to kill you. The land itself can also be hostile – marshes give off poisonous gases that can rapidly deplete your strength although there are healers in several of the towns who will cure you – for a price.

There are dungeons to be discovered and explored, and shrines to be sought out. Travel is usually on foot although you may be able to beg, steal or borrow horses to speed up your journey. You will certainly need a ship to visit the islands, not all of which are on your map and you will have to learn how to use the moongates efficiently. There is also rumoured to be a balloon or something similar hidden away.

As you travel the land and talk to the people, you can try and persuade them to







join your party. Up to seven others can join – one from each of the other profession types – paladin, bard, tinker and shepherd being the others. Some will only join you when you prove yourself worthy to a certain degree. Experience points are awarded for killing an enemy. Visit Lord British and he will promote you when you have gained sufficient points.

Magic plays a crucial part in the game. There are 26 spells to be mastered but before you can think about casting one, you need to know the ingredients and mix them in their correct proportions. There are six main ingredients that can be bought at any good herb shop but all the powerful spells require mandrake or nightshade which are not so easily come by. The spells range in power from simple ones, such as healing wounds or casting a magical light, through fireballs and assorted protections to kill and jinx – a jolly little trick that causes your opponents to attack themselves instead of you. Everybody except fighters and shepherds has some magical ability although how much depends on your job and experience level.

There is a lot of fighting to be done if you are to attain your quest. Combat takes place on a variety of tactical displays and you can move, attack or cast spells for each of your characters in turn. How the battle goes depends on what weapons and armour your party owns and how you deploy your forces. The use of slings and bows, especially by the members at the back of your party is recommended. The monsters – over 20 different types – fight intelligently and will run away if hard pressed leaving behind a treasure chest, although this is frequently trapped as you try to open it.

So what of the quest itself? The first part involves attaining a partial Avatarhood in the eight virtues – honour, valour, humility, sacrifice, honesty, compassion, justice and spirituality. The seer Hawkwind keeps you informed as to your progress and should be visited frequently. At the appropriate time, you have to go and meditate at a shrine – provided that you have found the correct rune to gain you admission and have learned the apposite mantra to chant. Only then will you be granted a vision. Apart from that, you will need to find some coloured stones – I found the red one on the eight level of Dungeon Destard. After that, I don't know apart from the fact that there will be some final conflict in a place known only as the Abyss.

Other things to look out for are secret passages (which abound in castles and dungeons) and the guild where you will need to purchase magical keys and gems (very useful for mapping dungeons). A sextant will also be an essential purchase – if you can find someone to sell you one. Above all, you gain information and write everything down. The amount of work and expense that you have to put into gaining even the smallest clue is phenomenal.

The display is in three main boxes. A large map displays your current position (line of sight vision only). The top right hand box displays the statistics for you and your party whilst the bottom box is used for command entry and as a general information box. Everything in the game is controlled by single keystroke commands apart from conversations which usually only require a single word. The game comes beautifully packaged with two large books, a map and a reference card.

Ultima IV is a superb game and streets ahead of any of its rivals. To date, I have played it for well over 60 hours and still feel that I have only scratched the surface of it. If you only buy one game this year, make sure it's Ultima IV.

Touchline

Ultima IV — C64
US Gold: Unit 10, The Parkway Industrial Estate, Heneage St, Birmingham B7 47Y
Price: £19.95 – two disks

TOP DRAW

Allen Webb explores the complexities of medium-res graphics.

Everyone rabbits on about high resolution graphics and how they can be used for brilliant effects and works of art. I am equally guilty having in the past discussed their use (January issue). For some applications, however, it may be possible to settle for a lower resulution system.

The C64, in common with most other micros, has a resident set of standard characters. Amongst these are a number which represent squares one quarter the size of a normal character. These squares can be used to plot lines or dots. Using this system you can achieve a resolution of 80 points across and 50 points up. Whilst you may not consider this too much of an idea, I recently saw two superb pictures drawn in this resolution. This emphasises that artistic ability can overcome system limitations. Remember, also, that Jeff Minter's excellent Psychedelia uses this sort of resolution.

The routines given here give complete control over the drawing of lines and dots and the manipulation of screen areas.

The commands have the following syntax:

1) Dots

SYS 51712,XP,YP, MODE, COLOUR

XP,YP are the co-ordinates of the dot.

MODE decides how the dot is drawn:

0 — erases the dot

1 — draws the dot

2 — flips the dot i.e. sets it if it's clear, clears it if it's set.

COLOUR specifies the colour of the dot. Values of zero to 15 change the colour. A value of 16 leaves the colour unaltered.

2) Lines

SYS 51715,X1,Y1,X2,Y2,MODE, COLOUR

X1,Y1,X2,Y2 are the co-ordinates at the ends of the lines.

3) Area manipulation

SYS 51718,XC,YC,WI,HI, COLOUR,MODE,[CHARACTER]

XC,YC specify the position of the top left hand corner of the area.

WI is the width of the area.

HI is the height of the area.

COLOUR acts in the same way as the previous commands.

MODE has the effects:

0 — EOR's the area i.e. changes it to reverse field. Repeating the command restores the area.

1 — fills the area with the character specified.

CHARACTER is only required if MODE equals one. A syntax error is generated if it is omitted when MODE=1 or if it is added when MODE=0. The character value is the POKE value so that a value of 32 erases the area and a value of one fills the area with the letter A.

This command acts on a 40 by 25 resolution and, as before, out of range values are ignored.

I've included a simple demonstration which shows some ways of using these commands. The first uses shades of grey and dotted lines to give a 3D effect. The second is just pretty and uses the area command to EOR the pattern.

MODE has the following effects:

0 — erases the line.

1 — draws the line.

2 — flips the line.

3 — draws a dotted line.

COLOUR is the same as for the dot command.

Both the dot and line commands use the 80 by 50 resolution with the origin in the bottom left corner of the screen. All out of range values are ignored.

PROGRAM: LOW RES LOADER

```
2000 FOR L=0 TO 56:CX=0
   :FOR D=0 TO 15:READ A
   :CX=CX+A:POKE 51712+L*16+
   D,A:NEXT D
2010 READ A:IF A<>CX THEN PR
   INT"ERROR IN LINE";
   2040+(L*10):STOP
2020 NEXT L:END
2040 DATA 76,9,202,76,99,204,
   76,193,204,32,12,203,141,
   135,3,32,1697
2050 DATA 12,203,141,134,3,
   32,12,203,141,136,3,201,2,
   240,6,169,1638
2060 DATA 1,56,237,136,3,141,
   136,3,32,12,203,141,137,3,
   72,152,1465
2070 DATA 72,138,72,169,0,
   141,132,3,141,133,3,133,
   252,173,134,3,1699
2080 DATA 133,251,201,50,144,
   3,76,247,202,173,135,3,
   201,80,144,3,2046
2090 DATA 76,247,202,169,49,
   56,229,251,133,251,78,135,
   3,46,133,3,2061
2100 DATA 70,251,46,133,3,6,
   251,6,251,6,251,165,251,6,
   251,38,1985
2110 DATA 252,6,251,38,252,
   24,101,251,133,251,133,
   253,165,252,109,136,2607
2120 DATA 2,133,252,174,133,
   3,169,1,141,133,3,224,0,
   240,6,14,1628
2130 DATA 133,3,202,144,246,
   172,135,3,177,251,162,0,
   221,252,202,240,2543
2140 DATA 8,232,224,16,144,
   246,76,247,202,173,136,3,
   201,1,240,20,2169
2150 DATA 201,2,240,8,138,13,
   133,3,24,170,144,21,138,
   77,133,3,1448
2160 DATA 24,170,144,13,173,
   133,3,73,255,141,133,3,
   138,45,133,3,1584
2170 DATA 170,189,252,202,
   172,135,3,145,251,56,169,
   216,237,136,2,141,2476
2180 DATA 138,3,165,251,133,
   253,24,165,252,109,138,3,
   133,254,173,137,2331
2190 DATA 3,201,16,240,2,145,
   253,104,104,168,104,96,32,
   126,123,97,1814
2200 DATA 124,226,255,236,
   108,127,98,252,225,251,
   254,160,32,253,174,32,2807
2210 DATA 138,173,32,247,183,
   165,20,96,72,152,72,138,
   72,169,1,141,1871
2220 DATA 136,3,173,139,3,
   201,80,144,3,76,247,202,
   173,140,3,201,1924
2230 DATA 80,144,3,76,247,
   202,173,141,3,201,50,144,
   3,76,247,202,1992
2240 DATA 173,142,3,201,50,
   144,3,76,247,202,173,140,
   3,56,237,139,1989
2250 DATA 3,141,143,3,173,
```

```
142,3,56,237,141,3,141,
    144,3,169,1,1503
2260 DATA 141,145,3,141,146,
    3,173,144,3,16,10,160,255,
    140,145,3,1628
2270 DATA 73,255,24,105,1,
    141,147,3,173,143,3,16,10,
    160,255,141,1650
2280 DATA 146,3,73,255,24,
    105,1,141,148,3,173,148,3,
    56,237,147,1663
2290 DATA 3,141,149,3,16,35,
    169,255,141,150,3,169,0,
    141,151,3,1529
2300 DATA 173,147,3,141,152,
    3,173,148,3,141,153,3,173,
    144,3,48,1608
2310 DATA 5,169,1,141,150,3,
    76,217,203,169,0,141,150,
    3,169,255,1852
2320 DATA 141,151,3,173,148,
    3,141,152,3,173,147,3,141,
    153,3,173,1708
2330 DATA 143,3,48,5,169,1,
    141,151,3,173,152,3,141,
    154,3,56,1346
2340 DATA 237,153,3,141,155,
    3,173,153,3,141,156,3,78,
    152,3,173,1727
2350 DATA 153,3,56,237,152,3,
    141,157,3,173,139,3,141,
    135,3,173,1672
2360 DATA 141,3,141,134,3,
    173,158,3,201,3,208,5,173,
    136,3,73,1558
2370 DATA 255,141,136,3,32,
    46,202,173,157,3,16,30,24,
    109,156,3,1486
2380 DATA 141,157,3,173,139,
    3,24,109,151,3,141,139,3,
    173,141,3,1503
2390 DATA 24,109,150,3,141,
    141,3,76,85,204,56,237,
    155,3,141,157,1685
2400 DATA 3,173,139,3,24,109,
    146,3,141,139,3,173,141,3,
    24,109,1333
2410 DATA 145,3,141,141,3,
    206,154,3,48,3,76,249,203,
    104,170,104,1753
2420 DATA 168,104,96,32,12,
    203,141,139,3,32,12,203,
    141,141,3,32,1462
2430 DATA 12,203,141,140,3,
    32,12,203,141,142,3,32,12,
    203,141,158,1578
2440 DATA 3,201,2,240,13,201,
    3,240,9,169,1,56,237,158,
    3,141,1677
2450 DATA 158,3,32,12,203,
    141,137,3,173,139,3,205,
    140,3,144,11,1507
2460 DATA 72,173,140,3,141,
    139,3,104,141,140,3,173,
    141,3,205,142,1723
2470 DATA 3,144,11,72,173,
    142,3,141,141,3,104,141,
    142,3,76,24,1323
2480 DATA 203,32,12,203,141,
    139,3,32,12,203,141,141,3,
    32,12,203,1512
2490 DATA 141,148,3,32,12,
    203,141,147,3,32,12,203,
    141,137,3,32,1390
2500 DATA 12,203,141,138,3,
    240,6,32,12,203,141,158,3,
    24,24,173,1513
2510 DATA 139,3,109,148,3,
    201,41,144,1,96,24,173,
    141,3,109,147,1482
2520 DATA 3,201,26,144,1,96,
    173,148,3,240,238,173,147,
    3,240,233,2069
2530 DATA 173,147,3,109,141,
    3,141,147,3,32,85,205,160,
    0,173,138,1660
2540 DATA 3,208,18,177,251,
    73,128,145,251,173,137,3,
    201,16,240,19,2043
2550 DATA 145,253,76,67,205,
    173,158,3,145,251,173,137,
    3,201,16,240,2246
2560 DATA 2,145,253,200,204,
    148,3,208,213,238,141,3,
    173,141,3,205,2280
2570 DATA 147,3,208,197,96,
    169,0,133,251,133,252,172,
    141,3,240,16,2161
2580 DATA 24,165,251,105,40,
    133,251,165,252,105,0,133,
    252,136,208,240,2460
2590 DATA 24,165,251,109,139,
    3,133,251,165,252,105,0,
    72,109,136,2,1916
2600 DATA 133,252,104,24,105,
    216,133,254,165,251,133,
    253,96,255,21,255,2650
```

PROGRAM: LOW RES DEMO

```
1 POKE 53281,11
2 C1=12:C2=15
10 SA=51712
20 PRINT CHR$(147)
21 FOR X=0 TO 49
22 SYS SA+3,X,0,X,20,3,C2
23 SYS SA+3,X,49,X,29,3,C2
25 NEXT
26 FOR X=50 TO 60
27 SYS SA+3,X,0,X,49,2,C2
   :NEXT
28 FOR X=61 TO 79
29 SYS SA+3,X,0,X,49,2,C1
   :NEXT
30 SYS SA+3,61,0,61,49,0,C2
35 Y1=0:Y2=49:DX=49
   :FOR X=0 TO 19
40 SYS SA+3,X,Y1,X,Y2,1,C2
50 SYS SA+3,X+DX,Y1,X+DX,Y2,
   1,C2
60 Y1=Y1+1:Y2=Y2-1:DX=DX-2
   :NEXT
70 FOR X=8 TO 11
80 SYS SA+3,X,0,X,49,0,0
90 SYS SA+3,X,12,X,37,1,C1
120 SYS SA+3,X,0,X,11,3,C2
125 SYS SA+3,X,39,X,49,3,C2
   :NEXT
130 FOR X=40 TO 47
140 SYS SA+3,X,0,X,49,0,0
150 SYS SA+3,X,10,X,39,1,C1
160 SYS SA+3,X,0,X,9,3,C2
165 SYS SA+3,X,41,X,49,3,C2
170 NEXT
175 FOR D=1 TO 1000:NEXT
   :POKE 53281,0:PRINT"
   [BLACK,CLEAR]"
180 PRINT"[CLEAR]"
190 FOR A=0 TO 2*πSTEP.1
200 XL=40+COS(A)*39
   :YL=25+SIN(A)*24
210 SYS SA+3,39,24,XL,YL,1,C
   :C=C+1
220 NEXT
230 FOR D=1 TO 1000:NEXT
240 FOR I=1 TO 30
   :XO=RND(1)*20:YO=RND(1)*10
   :WI=RND(1)*40:HI=RND(1)*25
250 SYS SA+6,XO,YO,WI,HI,16,0
   :NEXT
260 FOR Y=0 TO 39
270 SYS SA+6,0,Y,39,1,16,0
   :NEXT:GOTO 240
```

WELCOME TO THE MACHINE

Allen Webb takes you one step further in your quest to master machine code.

LAST MONTH WE STARTED TO look at the various ways of movinv data about. Whilst we concentrated on the screen, the principles apply equally to the movement of data anywhere. I hope you found the homework easy. With the material we covered last time, you should be able to come up with two or three solutions to each problem.

First, I wanted a routine to put a row of stars along the top of the screen. Here is one solution:

```
10  ASSEMBLE 100, I
100 REM *=$C000
110 REM LDA £19
120 REM JSR $FFD2
130 REM LDY £0
140 REM LDA £42
150 REM .LOOP: JSR $FFD2
160 REM INY
170 REM CPY £40
180 REM BNE LOOP
190 REM RTS
200 REM ]
```

This uses the print character routine in ROM to print asterisks to the screen at the current cursor position. Lines 110 and 120 print HOME (CHR$(19)). I used this method since there is no need to worry about updating the colour matrix. Those of you with old ROM 64s will no doubt be aware that when you set the cursor colour, the colour matrix is not updated. That means that if you move data direct to the screen memory, you will not necessarily get the colours you want. New ROM 64s have had this fixed.

The second problem asked you to print the character set on the screen. Here is my solution:

```
10  ASSEMBLE 100,I
100 REM *=$C000
110 REM LDY £0
120 REM .LOOP: TYA
130 REM STA £0400, Y
140 REM LDA £1
150 REM STA $D800, Y
160 REM INY
170 REM BNE LOOP
180 REM RTS
190 REM ]
```

In this routine, I have used simple indexing to put characters at the start of the screen memory. Since I don't want to change the address indexed, indirect indexing is unnecessary. There are 256 characters with POKE values ranging from zero to 255. I use this fact in line 120 by using the Y register to update the character to be POKEd. Lines 140 and 150 take care of the colour matrix for old ROM routines. The Basic equivalent to this routine is:

```
10 FOR I = 0 to 255
20 POKE 1024+I, I
30 POKE 55296,1
40 NEXT I
```

Line 170 acts in a slightly different way to the looping we've used previously, but I'll cover that shortly.

There is one more addressing mode which you should be aware of. This is an infrequently used mode called Pre-Indexed Indirect Addressing. This mode uses the X register to look for an address in a table and act on the address. The mnemonic for this mode has the form:

(address,X)

where address is a zero page location. Here are some examples:

```
LDA ($FB,X)
STA ($SAA,X)
```

Its operation takes a little understanding, but here is what it does. Imagine that you have a table of 16 bit addresses stored as a table in zero page starting at $AA.

$AA low byte address 1
$AB high byte address 1
$AC low byte address 2
$AD high byte address 2
$AE low byte address 3
$AF high byte address 3
and so on...

If X contains the value zero, the instruction LDA ($AA,X) does the following:
1) Adds the contents of X (i.e. zero) to the address $AA to give $AA.
2) the accumulator is loaded by the contents of the address held in the resulting byte pair $AA and $AB.

Similarly, if X contains two, then the accumulator will be loaded with the contents of the address pointed at by $AE,$AF.

This is not an addressing mode that you will use often, but it's worth knowing about, in case you have a need for it one day.

Last month, I introduced the use of conditional branchng. At that time, it was simply to allow us to make progress and I made no attempt to discuss it at length. It is now necessary to look at it in some depth.

In the microprocessor is a register called the Status Register. This eight bit register is used to hold seven flags, each using one bit. The flags held are as follows:

i) The Carry Flag (C)
This flag is used to carry information on which arithmetic operations are performed. If, for example, two numbers are added to give a result greater than 256, the carry flag is set so that you can take appropriate action. We'll discuss this when I deal with 16 bit arithmetic.

ii) The Overflow Flag (V)
Only the first seven bits are used for holding data, the eighth being a sign bit. Hence only numbers in the range +127 to −127 are used. If an operation attempts to store greater than +127, then the overflow flag is set. Again, we'll discuss at a later date.

iii) The Negative Flag (N)
This is set if an operation results in a negative answer.

iv) The Decimal Flag (D)
This is set if you wish to work in decimal (BCD) mode.

vi) The Interrupt Flag (I)
Set if an interrupt is in progress.

vii) The Zero Flag (Z)
Set if an arithmetic operation gives a zero result.

The branch instructions test the status of a flag and act accordingly. The instructions provided are:

BCS — branch if carry flag is set
BCC — branch if carry flag is clear
BEQ — branch if zero flag is set
BNE — branch if zero flag is clear
BMI — branch if negative flag is set
BPL — branch if negative flag is clear
BVC — branch if overflow flag is clear
BVS — branch if overflow flag is set

You will generally use these instructions directly after an arithmetic operation. The most usual are:

CMP — this compares the accumulator to data or the contents of a location.
CPY — compares the Y register to data.
CPX — X analogue to CPY.

These three instructions perform a non-destructive comparison by subtracting the data from the register and updating the status flag accordingly depending on whether the result is zero, positive or negative.

Register = Data — set carry flag
Register = Data — set zero flag
Negative register is changed by the sign bit.

So to detect various results, you use:

Register = data — use BCS, e.g.
CMP £4
BCS LOOP
branches of LOOP if accumulator holds four or more.

Register data — use BCC, e.g.
CPY £10
BCC LOOP
branches to LOOP if Y register holds less than 10.

Register = data — use BEQ. e.g.
CPX £6
BEQ LOOP
branches if X register holds six.
Register data — use BNE, e.g.
CMP £3
BNE LOOP
branches if accumulator does not hold three.

If you now look back at last month's examples you will see how these tests are used. Arithmetic instructions such as INY,INX,INC,DEY,DEX,DEC change the negative and zero flags depending on the result. (INC and DEC increment and decrement a memory location by one.)

My answer to question two from last month's homework uses this effect. Line 160 increments the counter. When it reaches 255, adding one more will result in zero. Since this signifies that we have finished, I use BEQ in line 170 to detect this situation.

Now we've collected together the basic tools, let's start writing some decent routines. In the last part, we discussed the use of NOP instructions to create delays. To achieve more substantial pauses we need to use more complex routines. Here is a simple delay routine:

```
100 ASSEMBLE 110,I
110 REM *=$C000
120 REM LDX £$10
130 REM .LOOP1: LDY £$10
140 REM .LOOP2: DEY
150 REM BNE LOOP2
160 REM DEX
170 REM BNE LOOP1
180 REM RTS
```

This routine uses a pair of nested loops to wait a short time. The values loaded into the X and Y registers in lines 120 and 130 decide the delay. Lines 140 and 150 count down the Y register to zero. This process is then repeated the number of times in the X register. The Basic equivalent of this routine would be a pair of nested loops such as:

```
FOR X=0 TO 10: FOR Y=0 TO 10: NEXT Y,X
```

An alternative method is to call the routine at $EEB3. This routine generates a one millisecond delay.

Let's use this delay routine to generate a synapse tweaking pattern. Consider the routine:

```
100 ASSEMBLE 110,I
110 REM *=$C000
120 REM LDA £6
130 REM STA 998
140 REM LDA £4
150 REM STA 999
160 REM .LOOP: INC $D020
170 REM INC 1000
180 REM JSR DELAY
190 REM JSR $FFE1
200 REM BEQ FINISH
210 REM JMP LOOP
220 REM .FINISH: RTS
230 REM .DELAY: LDX 998
240 REM .LOOP2: LDY 999
250 REM .LOOP1: DEY
260 REM BNE LOOP1
270 REM DEX
280 REM BNE LOOP2
290 REM RTS
300 REM ]
```

You'll immediately recognise lines 250 onwards as being our delay routine. The delay parameters are held in locations 998 and 999 rather than being loaded as direct values. The line 120 to 150 set up the delay parameters. The core of the routine is lines 160 to 210. It is an infinite loop which changes the colour of the border, delays a bit and then loops back. The subroutine call in line 19 tests the RUN/STOP key. If this key is pressed, then the Z flag is set. Line 200 checks this and stops if the flag is set. Try messing about with the delay values and see the effect. If you use a bit of care and possibly the odd NOP to fine tune it, you may be able to get some starionary coloured bands in the border.

The next, and last example, is a little more useful:

```
100 ASSEMBLE 110,I
110 REM *=$C000
120 REM LDA £$A0
130 REM STA 998
140 REM LDA £$30
150 REM STA 999
160 REM LDY £0
170 REM .LOOP3: LDA £42
180 REM STA $0400,Y
190 REM LDA £1
200 REM STA $D800,Y
210 REM TYA
220 REM JSR DELAY
230 REM TAY
240 REM LDA £32
250 REM STA $0400,Y
260 REM TYA
270 REM JSR DELAY
280 REM TAY
290 REM LDA TABLE,Y
300 REM BEQ FINISH
310 REM STA $0400,Y
320 REM TYA
330 REM JSR DELAY
340 REM TAY
350 REM INY
360 REM BNE LOOP3
370 REM .FINISH: RTS
380 REM .TABLE: £B:25,15,21,
18,32,3,15,13,13,15,4,15,18,5,0
390 REM .DELAY: LDX 998
400 REM .LOOP2: LDY 999
410 REM .LOOP1: DEY
420 REM BNE LOOP1
430 REM DEX
440 REM BNE LOOP2
450 REM RTS
460 REM ]
```

Again the delay routine uses two locations to hold the parameters. This routine simulates a device rather like the "vidiprinter" used on the Saturday afternoon football results service on TV. A message is slowly printed across the screen with a flashing asterisk cursor. The routine is quite simple.

Line 160 zeros the Y register which will act as our counter. Lines 170 and 180 print an asterisk in the top left hand corner of the screen. Lines 190 and 200 update the colour matrix for you folks with old ROM machines. The next three lines force a delay. The TYA and TAY either side of the call to the delay loop save the contents of the Y register since it is used in the delay. Lines 240 and 250 erase the asterisk with a space and we wait a while longer. Finally, lines 290 and 310 take a letter from the table and put it on the screen. Line 300 checks for a zero value in the table. This is used to mark the end of the table so that the routine stops at the end of the message. Line 350 increments the counter and provided that we don't go over a value of 255, line 360 sends us back for the next character.

I realise that I'm spending a lot of time explaining how the routines work. As we progress, I will make briefer comments since you should soon be able to suss things out for yourselves.

OK, homework time. First, I want a routine which will fill the entire screen with a specified character. I don't expect the best solution but I've told you enough for a crude but effective routine.

Secondly, I want a routine which will more a block of data from the top line of the screen to, say, the 20th line. A single line of data will suffice but you can easily move u to 256 bytes. This sort of routine is frequently used in a range of situations.

Finally, how about a routine which will scroll the top line of the screen one step to the right with the leftmost character replaced with a space?

Next month we'll explore eight and 16 unsigned arithmetic.

Teacher's Pet

Margaret Webb browses through some readily available educational software.

The supply of new educational software seems to have dried up, so I decided to look around the local shops to see what was on offer. The answer seems to be, very little. The reasons for this could be threefold:

1. My hometown is poor for shops selling software.
2. The storeowners are very cautious about stocking educational software.
3. There may genuinely be little software.

I suspect that notwithstanding the claims of the value of computers for education, the reality is that education is poor business. Much more money can be made by selling games. The majority of games only require good programming whilst educational material requires detailed teaching knowledge as well as programming ability. These rather disappointing facts of life were only too evident at the last PCW show where a number of exhibitors stated that they were no longer interested in educational software since it had no future. This is all rather sad since quite a lot of the important pre-school spade work can be carried out with computer assistance with the software acting as a type of expert system (parents aren't always teachers). For the older children, software can be used to provide revision material and to support conventional didactic methods.

Whilst rummaging through the shops, however, I did see an interesting item. Not a new product, this is a triple pack of Hill MacGibbon software for a touch less than the original price of one. Hill MacGibbon is an interesting company in that it has produced software for most of the popular computers. In some of these packages there has been collaboration with well known companies such as Collins and Pan.

In light of this, this is a good time to take a look at which packages are available for the C64. The triple pack contains Ballooning, Car Journey and Secret Agent. Between them they provide quite a comprehensive package, each coming with a colour booklet dealing with diverse aspects of the topic and ideas for further work.

Ballooning

The title is self explanatory. The booklet deals with the historical aspects of ballooning and then goes on to look at the Hindenburg disaster and how a hot air balloon works. The ability to read and use a map is very important when flying a balloon and this topic is also covered. The software provides a balloon simulator with controls showing your altitude, fuel (it's a hot air balloon), rate of climb and atmospheric temperature. Using the information given on the instruments, you must guide the balloon over varying terrain. In this way, ballooning covers aspects such as physics, map-reading, mathematics and geography.

Car Journey

This would appear to be a misnomer since the software involves the operation of a light haulage business. In it you must run the firm and keep it financially viable. This is done by judiciously securing contracts to move goods from point to point and selecting the optimum routes. You get bonuses or penalties depending on whether to keep to the time limits. Naturally, you must attempt to choose contracts which end up at the starting points of other contracts. It doesn't pay to run an empty truck! Different size vehicles are available to suit different size consignments.

The accompanying booklet deals with aspects of the car, how it works and its history. It also covers the history of roads and transport and the motorway system. A nice touch is an extract from Toad of Toad Hall describing Toad's discovery of the joys of motoring. The pack covers reading, mathematics, geography and mechanics.

Special Agent

This package puts you in the shoes of a budding James Bond searching Europe for the dastardly enemy agent who's gaily killing off your operatives. The game centres around a map of Europe showing the major cities. From time to time intelligence reports flash up at the foot of the screen. Some of these are in code presenting additional problems. You must act on the received information and travel from city to city. You must choose your trains and planes from timetables.

As usual, the booklet covers subjects connected with the central theme. You are introduced to the elements of cryptography, the capital cities of Europe and a little information on real life spies is given. It covers geography, reading, lateral thought and some mathematics.

Hill MacGibbon also offers a number of packages covering more specific subjects. Teaching the mechanics of reading isn't that much of a problem since children generally soon grasp the form of words. What is more difficult to grasp and contributes most to the subtleties of English is punctuation. Punctuation Pete is a program intended to help in this area. The child is presented with a graded piece of text which has had all punctuation and capital letters removed. The child must read the text and attempt to punctuate it so that the meaning is clear. When he feels that all is finished, the program marks the result showing any mistakes. Surprisingly, finding the correct punctuation is quite tricky.

Technically, the program is slick with large legible text and the use of an animated man as the cursor. I highly recommend it.

The last two programs are for the younger child. First, there is Picture Builder. As the name suggests, this program allows the manipulation of basic shapes such as squares, circles and triangles for the creation of pictures. The shapes can be stretched, shrunk, rotated and painted to give the required effect. Multicolour mode is used to allow up to four colours. For those of you with printers, there is a hard copy facility — nicely done and easy to use.

Finally, we have Run Rabbit Run. This is a simple game played on a matrix of squares, rather like a board game. You must guide a number of rabbits to their homes and away from the hungry fox. The game tells you how many squares your rabbits may jump on each move. These jumps must be distributed between the rabbits. There are bushes in which the rabbits can hide, if the fox isn't there first. The game ends when all rabbits are home or have been eaten.

This game is quite tough and forces the child to use a little logic, lateral thought and some counting skills.

The impressive features about Hill MacGibbon software are that a high standard of programming is used and there is a decent level of content. Unlike some educational material, there is evidence of real teaching input.

Touchline

Collins Soft/Hill MacGibbon: 7 Grafton St, London W1X 3LA Tel: 01 493 7070.

P Green brings you some hints on saving time and memory.

DATA, TIME AND MEMORY

TIME AND MEMORY CAN BE wasted when a Basic program contains a lot of numerical data statements. This article suggests ways in which you can save both, even on programs copied from magazines.

The Facts and Figures

If you have a lot of numerical data to be stored, there is a fairly quick and easy way to save time and memory. Save the block of memory straight on to the disk or tape and get the Basic program to load this data straight into memory instead of POKEing it in.

Time is saved by this method because when you use data statements within a program, you must first load the data in Basic form and then run the program to POKE the data into memory. This takes 12 seconds for the first half of the operation and six for the second half for each 1K of memory when using a disk drive. Alternatively, to transfer the data straight into memory from disk takes only five seconds for each 1K of memory. Of course, the saving is much greater if you are using tape since tape loading is a much more lengthy process.

Memory can be saved, both in the computer and on the disk or tape. In the computer, 1K of directly entered data, of course, takes up just 1K of memory. On the disk or tape, it takes up just over 1K of storage space. On the disk, this is five blocks. In the case of Basic data statements, besides occupying the 1K of memory once the program has been run, the Basic data also takes up memory – just under 3K. Altogether this method actually uses just under 4K of memory. On the disk or tape, it takes up well over 3K of storage space – that is, 15 blocks on the disk.

Saving the Data

So, firstly, how can we save a block of memory? And secondly, how can we get our Basic program to load it again?

This can be done in two ways. You can use a machine code monitor to save an area of memory. To do this you will have to get the start address of the block of memory that you wish to save and then calculate the end address and add one.

Your machine code monitor, of course, must not occupy the same area as the memory which you wish to save.

The procedure should be as follows:

1. Load and run the Basic program, or at least the part which POKEs the data into memory.
2. New the Basic program and load the machine code monitor.
3. On most machine code monitors, the instructions to save a block of memory is something like:

```
S "file name",C000,C200,08
```

This will save the block of memory from $C000 (the start address) to $C1FF (the end address) on a disk drive with the device number of 8 or 08 hex (01 for tape) with a filename "SPRITES". (You choose the filename although you do not actually need one for files saved on tape).

4. You may need to save more than one block of data, for instance, a block of character data at 12288 and a block of machine code data at 49152. All you need to do for this is to repeat instruction three for the new block of data, bearing in mind that for disks you will need a different filename.

Another and possibly easier method is to alter certain pointers in the zero page of memory so that you fool the computer into thinking that the block that you wish to save is a Basic program. The locations to note are 43 to 46 inclusive. Locations 43 and 44 are the low and high bytes of the start of Basic which are normally one and eight respectively. (256*8+1 = 2049 = start of normal Basic area). Locations 45 and 46 are the low and high bytes of the start of variables which is normally the end of the Basic program Plus one.

The procedure is as follows:

1. Calculate the start and end of the block of memory which you wish to save, not forgetting to add one to the end.
2. Calculate the high and low bytes from these figures in decimal.
3. Run the Basic program, or the section of it that POKEs the data into memory.
4. In direct mode, POKE in these figures into locations 43 to 46.
5. Save the block of memory by typing in SAVE "filename",8,1 or SAVE "filename",1,1
6. Repeat this for other blocks of memory if necessary.

The Basic Loader

Now that you have saved your block of memory, how do you get your program to load it again automatically? There are two ways in which you can do this. The first is to write a short machine code routine to load the blocks of memory. The second is to add one or more lines to the beginning of your Basic program. The machine code method requires an understanding of how the KERNAL load routine works and would take too long to describe here. Instead, I will concentrate on describing the Basic method.

It is actually very easy to get your Basic program to do the loading for you. The most important thing to remember is that once the first block of memory has been loaded, the program will start again from the beginning and if you do not do something to prevent it, the same block will be loaded again and again.

The first thing to do is save a copy of the full program, data statements and all, in case somthing goes wrong, and keep it safe. Next, remove the data statements and the READ-POKE routine and any error traps, and save the program again.

The first line of the program can be used to load the block of memory by using a line such as:

```
10 X=X+1 : IFX=1THENLOAD "filename",8,1
```

This filename is the same as the one you used to save the block of memory. If a tape is used, then first change the eight to a one, and then, don't use a filename. The figure one after the eight or one is required so that the block of memory goes back to whence it came.

If there is more than one block of memory to load, another line needs to be added, as follows:

```
20 IFX=2THENLOAD"2nd filename",8,1 etc.
```

or, in the case of tape, the first line can become:

```
10 X=X+1 : IFX 3THENLOAD"",1,1
```

Do not use a filename. This will load the first two blocks of memory found on the tape.

The way in which the loader works, is as follows:

1. When the program is run, X=0.
2. At line 1, X becomes 1 and the first load takes place.
3. After the load, the program starts again at line 10 but the variable X is still 1. Therefore X becomes 2 and since this is not equal to 1, the program continues to the next line.
4. This will go on until all the blocks of memory have been loaded and the rest of the program can continue.

FONT FACTORY

Evelyn Mills looks at a new product from Impex.

THE FONT FACTORY (FF) IS APTLY named and works hard for you, doing overtime at your request!

Firstly the requirements are a disk drive, printer and word processor. The printer should, for preference, be the Commodore Vic 1525/MPS 801 although directions are given for using a printer interface emulating the MPS 801 or 803. It is claimed that FF will work with most word processors with open sequential files and I have used Easyscript throughout with no problems; however it would be worthwhile doing a double-check with the distributors before purchase if you have another word processor as some do not link up.

Noticeably there is no manual supplied with FF; instead the program gets to work right away printing out full instructions using the directions given. The resulting 16 page manual is in two parts – one for Font Factory and one for Signwriter 64. Both are very well written and the full concepts of the programs are easily understood – no hidden complaints here!

Before using FF, create a file document with your word processor and save this to disk. There is no necessity to use the commands of your processor other than direct typing mode. However – and this is most important – your MUST enter 'θfni' at the beginning followed by (return). Should you wish to use a different font in the middle of your document, insert a new 'θfn2' header, followed by (return). FF has eight in-built fonts with which to play around.

Having saved your file to disk, load FF and let it take over. Initially I suggest that you use option three to print your document (there are plenty of screen instructions to help you along). Essentially FF will ask you to define your first font by selection from a list of eight; this will then be processed for you. The second font style will then be requested and FF gets back to work. When you have defined the number of fonts in your document, a simple (return) will exit you to the next option. If you select the parameters given on the screen (a good idea initially) insert your document when told to do so and FF will print your letter in the fonts selected, very simple indeed and very effective.

There are eight in built fonts including Micro, Bold, Roman, Gothic and, most important of all, the Descender. The latter gives you 'true type face' of a high quality as its name implies.

FF is full of options using normal or double width letters and has a very comprehensive list of embedded commands for centering, setting line width, left and right margins, optional page numbering and line spacing. All these commands are speedily screen controlled. Fonts may also be changed within your document (did I say versatile?).

More to come. You may define your own fonts, if desired, or edit existing ones. The whole process is extremely easy to use and 15 fonts may then be accessed at any one time within your document, including the in built fonts. Instructions are clear, concise and readily handled.

In effect you can create an entire character set or change characters from an existing set; if you do not like the A in Gothic font then change it! If you want to design the Greek alphabet – do it!

FF also has a signwriter program which may be loaded independently. Here again instructions are readily handled, when complemented with the manual. This functions independently of a word processor, character widths are Normal, Skinny or Double width and the output has two options; one selects print according to the printing characters in your slogan while the other selects solid line printing (note – the word cat in normal width prints a banner around two feet long!) F5 will stop the printer if you have been too enthusiastic.

As in FF, fonts may be changed and stored on disk. There is one Standard font in signwriter.

I see no problems in this program, consider it excellent value for money and doubt if you will be disappointed with its performance.

A really professional tool, agreeably prices and certainly "user friendly".

Scratchpad

Our readers provide more handy routines to add to your collection.

This month we are pleased to print three short utility programs. The First of these comes from Steven Freeman from Orpington and is a list utility.

How often have you tried to list a program on the screen of your C64 only to have the line that you want scroll off the top of the screen before you can read it? OK so you can slow this listing down with the CTRL key but the program still scrolls. Steven's handy routine alters the speed of the LIST command by changing the list vectors so that it jumps to a short machine code routine in spare memory (locations 679 to 702). Line 30 of the Basic loader contains the POKE that alters the speed of the list, you can change this if required. The other controls are:

f1 To pause the listing
f3 To slow down the listing

For all of those people who have stored machine code programs on tape only to forget where they load in memory, Mr K Peppin from Little Neston has provided an extremely handy Tape Header Reader program. The routine is very handy for locating a machine code program and giving its length.

This routine is reliant on a machine code call to the Kernal ROM which searches for any header to a program. The information is then stored in

PROGRAM: CATALOG

```
5 REM PROGRAM FOR THE PLUS/4
10 S$="(CLR)(RVS)(RED) *** MASTER TAPE PROGRAM CATALOGUE ***  (OFF)(BLK)"
15 PRINTS$
20 PRINT:READX:FORI=1TOX:READA$:PRINT" "A$:NEXTI
25 PRINT"(DOWN)WHICH PROGRAM ?":PRINT"(DOWN)ENTER THE PROGRAM NUMBER":INPUTA
30 IFA<1ORA>XTHEN25
35 POKE0,7
40 PRINTC$;"(DOWN)(DOWN)PRESS THE FAST FORWARD BUTTON":PRINT"(DOWN)THEN PRESS ANY KEY"
45 POKE239,0
50 IFPEEK(239)=0THEN50
55 IFPEEK(64784)=255THEN40
60 PRINT"(DOWN)PLEASE WAIT FOR PROGRAM TO BE FOUND"
65 POKE0,15:TI$="000000":B=(A-1)*10
70 IFTI/60<BTHEN70
75 POKE0,7
80 PRINT"(CLR)":PRINTS$:PRINT"(DOWN)(DOWN)PRESS THE STOP BUTTON ON THE CASSETTE"
85 VOL6:SOUND1,850,20:FORT=1TO90:NEXTT:SOUND1,800,20
90 IF PEEK(64784)=255THEN100
95 GOTO 90
100 PRINTS$:PRINT"(DOWN)YOU ARE NOW READY TO ":PRINT"(DOWN)LOAD/SAVE "
105 RESTORE:FORI=1TOA+1:NEXTT:READA$:NEXT
110 PRINTA$;:POKE0,15
115 POKE239,0
120 DATA 10:REM NUMBER OF PROGRAMS
125 DATA"PROGRAM 1........................."
130 DATA"PROGRAM 2........................."
135 DATA"PROGRAM 3........................."
140 DATA"PROGRAM 4........................."
145 DATA"PROGRAM 5........................."
150 DATA"PROGRAM 6........................."
155 DATA"PROGRAM 7........................."
160 DATA"PROGRAM 8........................."
165 DATA"PROGRAM 9........................."
170 DATA"PROGRAM 10........................"
```

the cassette buffer, address 828-1019, where it can be easily used.

The final routine in this month's Scratchpad is a very handy cassette tape catalogue system from N V Newen from Oxford. The object of the program is to allow easy access to either programs or subroutines stored on a master cassette.

The program as it stands will store 10 sections of approx 8K in length on a C60 cassette but this could quite easily be changed to suit individual requirements. If you use the C-16/Plus/4 fast tape routine from our Feb '86 issue about 10 programs of 25K can be saved. The data can be altered to suit the names of your programs. Don't forget to put the program number at the start of each line as it is this that you will have to press to access that file.

PROGRAM: LIST

```
10 REM 'SLOW LIST'[SPC2]STEVEN FREEMAN 1986
20 REM [F1] = PAUSE : [F3] = SLOW DOWN.
30 POKE 251,248 : REM SPEED OF LIST
40 FOR CODE=0 TO 22:READ MC:CS=CS+MC:POKE 679+CODE,MC:
NEXT CODE
45 IF CS<>3262 THEN PRINT"DATA ERROR!":END
50 POKE 774,167:POKE 775,2
60 DATA 72,165,197,201,4,240,250,201,5
70 DATA 208,8,165,251,133,162,165,162
80 DATA 48,252,104,76,26,167
```

PROGRAM: TAPE HEADER

```
10 PRINT"[CLEAR]":POKE 53280,0:POKE 53281,0
20 GOSUB 270
30 PRINT"[DOWN6,RIGHT5]INSERT TAPE AND PRESS A KEY."
40 GET A$:IF A$=""THEN 40
50 PRINT"[CLEAR]"
60 SYS 63276
70 PRINT"[CLEAR]"
80 GOSUB 270
90 POKE 646,14:PRINT"[DOWN2,RIGHT]PROGRAM NAME:";
100 POKE 646,2:FOR A=833 TO 848
110 PRINT CHR$(PEEK(A));:NEXT
120 PRINT
130 X=(PEEK(830)*256)+PEEK(829)
140 Y=(PEEK(832)*256)+PEEK(831)
150 POKE 646,14:PRINT"[DOWN2,RIGHT]START ADDRESS:";
160 POKE 646,2:PRINT X
170 POKE 646,14:PRINT"[DOWN2,RIGHT]END ADDRESS:";
180 POKE 646,2:PRINT Y
190 POKE 646,14:PRINT"[DOWN2,RIGHT]LENGTH:";
200 POKE 646,2:PRINT Y-X;:POKE 646,14:PRINT"BYTES."
210 PRINT"[DOWN2,RIGHT]TYPE:";:POKE 646,2
220 IF PEEK(828)=1 THEN PRINT"[RIGHT]BASIC"
230 IF PEEK(828)=3 THEN PRINT "[RIGHT]MACHINE CODE"
240 POKE 646,14:PRINT"[DOWN2,RIGHT12]PRESS ANY KEY"
250 GET A$:IF A$=""THEN 250
260 GOTO 10
270 PRINT"[RIGHT10]TAPE HEADER READER"
280 PRINT"[DOWN,RIGHT17]BY"
290 PRINT"[DOWN,RIGHT13]K.PEPPIN 1985"
300 RETURN
```



FROGGY

Daryl Bowers shows you how to kill the frog!

WELCOME BACK TO THE saga of the Frog. This month we will be adding the routine which controls the status panel at the bottom of the screen, and the routine which kills the Frog. Although the latter will not function until the checking routines are added in the next issue.

Info

The routine begins by printing the value of LIVES on to the status panel. ONLY indicates whether the frog is dead or alive and if dead (ONLY = 1) then we exit from the routine.

The next four instructions print the values in FOOD and FOOD+1 and the following seven use the X register as an index to point to the five digits of METRES and SCORE and to place them correctly on the screen.

One of the features of the game is that it becomes more difficult as you progress, and this is achieved by increasing the speed of the BIRD and the FLY. Their speed is increased by a small amount every time another 100 points is scored. The next two lines store the current digit in the 100 column of the score into NXTHUN. The code which carries out the increase in speed appears later in the routine.

Lines 10010 to 10100 decrease the two byte delay METDEL and METDEL2 to check whether another metre has passed. If not then we jump to FOD, which simply returns from the routine.

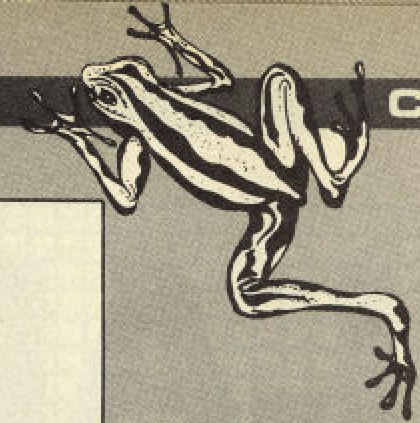
The next eight lines use the X register as an index to increase the METRES value. If the digit being increased reaches nine-plus-one in value, (i.e. 10), then that digit is set to zero, and the next highest digit is increased. METSPD contains the value which is placed back into

```
1430 TABMOV     .WORD   MOVE1,MOVE4,MOVE1,MOVE5,MOVE6,MOVE2

1520            JSR PLAYER

1560            JSR INFO

9790            ;
9800            ;
9810 INFO
9820            ;
9830            ;
9840            ;
9850            LDA LIVES
9860            STA $0769
9870            LDA ONLY
9880            BEQ NOTONLY
9890            RTS
9900 NOTONLY
9910            ;
9920            LDA FOOD
9930            STA $07B9
9940            LDA FOOD+1
9950            STA $07BA
9960            ;
9970            LDX #4
9980 LOOP23     LDA METERS,X
9990            STA $0756,X
10000           LDA SCORE,X
10010           STA $07A6,X
10020           DEX
10030           BPL LOOP23
10040           ;
10050           LDA SCORE+2
10060           STA NXTHUN
10070           DEC METDEL
10080           BNE FOD
10090           DEC METDEL2
10100           BNE FOD
10110           LDX #4
10120 LOOP24    INC METERS,X
10130           LDA METERS,X
10140           CMP #'9'+1
10150           BNE OK23
10160           LDA #'0'
10170           STA METERS,X
10180           DEX
10190           BPL LOOP24
10200           RTS
10210           ;
10220 OK23
10230           ;
10240           LDA METSPD
10250           STA METDEL2
10260           ;
10270           DEC FOOD+1
10280           LDA FOOD+1
10290           CMP #'0'-1
10300           BNE SCOR
10310           LDA #'9'
10320           STA FOOD+1
10330           DEC FOOD
10340           LDA FOOD
10350           CMP #'0'-1
10360           BNE SCOR
10370           LDA #1
10380           STA DED
10390           ;
10400           ;
10410 SCOR
10420           ;
10430           LDX #4
10440 LOOP25    INC SCORE,X
10450           LDA SCORE,X
10460           CMP #'9'+1
10470           BNE OK24
10480           LDA #'0'
10490           STA SCORE,X
10500           DEX
10510           BPL LOOP25
10520           RTS
10530           ;
10540 OK24
10550           LDA SCORE+2
10560           CMP NXTHUN
10570           BEQ NOINC
10580           DEC FLYSPD
10590           DEC BRDSPD
10600 NOINC
10610           LDY #1
10620           LDA BRDSPD
10630           BNE OK33
10640           STY BRDSPD
10650 OK33
10660           LDA FLYSPD
10670           BNE OK34
10680           STY FLYSPD
```



```
10690 OK34
10700 FOD       RTS
10710           ;
10720           ;
10730           ;
10740           ;
10750           ;
10760 DEAD
10770           ;
10780           LDA DED
10790           BNE DEADED
10800           RTS
10810           ;
10820 DEADED    LDX #217
10830           STX $07F9
10840           INX
10850           STX $07F8
10860           LDY STAGE
10870           LDA JUMPTYPE
10880           BNE BIG3
10890           LDA XTAB1,Y
10900 BACK      SEC
10910           SBC #48
10920           STA $D002
10930           JMP YPOS
10940 BIG3
10950           LDA XTAB1B,Y
10960           JMP BACK
10970 YPOS
10980           LDA #158
10990           STA $D001
11000           STA $D003
11010           ;
11020           LDA #8
11030           STA $D020
11040           STA $D021
11050           ;
11060           ;
11070           ;
11080           SEI
11090           LDA #FININT&255
11100           STA $0314
11110           LDA #FININT/256
11120           STA $0315
11130           DEC LIVES
11140           JSR WAIT2
11150           LDA #1
11160           STA ONLY
11170           JSR INFO
11180           LDA #0
11190           STA ONLY
11200           JSR WAIT3
11210           LDA LIVES
11220           CMP #'0'
11230           BEQ FINAL
11240           LDA #HANDLE&255
11250           STA $0314
11260           LDA #HANDLE/256
11270           STA $0315
11280           LDA #'9'
11290           STA FOOD
11300           STA FOOD+1
11310           LDA #0
11320           STA DED
11330           LDA #$FF
11340           STA FLYXLO
11350           STA BRDXLO
11360           LDA #1
11370           STA FLYXHI
11380           STA BRDXHI
11390           LDX #3
11400 LOOP32
11410           LDA FRENCH2,X
11420           STA FRENCH,X
11430           DEX
11440           BPL LOOP32
11450           CLI
11460           RTS
11470           ;
11480           ;
11490           ;
11500           ;
11510           ;
11520           ;
11530           ;
11540           ;
11550 WAIT2
11560           LDA #5
11570 LOOP29    JSR BIGGY
11580           SEC
11590           SBC #1
11600           BNE LOOP29
11610           RTS
11620           ;
11630 WAIT3
11640           LDA #10
11650 LOOP30    JSR BIGGY
11660           SEC
11670           SBC #1
11680           BNE LOOP30
11690           RTS
11700           ;
11710 BIGGY
11720           LDX #$FF
11730           LDY #$FF
11740 LOOP28
11750           DEY
11760           BNE LOOP28
11770           DEX
11780           BNE LOOP28
11790 NOTDEAD   RTS
11800 FINAL     JSR WAIT3
11810           ;
11820           JSR INIT
11830           RTS
11840           ;
11850 FINISH
11860           .END
```

METDEL2, so by changing this value (line 1360) you can adjust the distance that must be travelled before the value of METRES is increased.

The same principle used for increasing the metres is used next to decrease FOOD and increase SCORE.

Which brings us to OK24. If the value of NXTHUN is different to the current value of SCORE+2, then it is time to decrease the delays which control the speeds of the BIRD and FLY, BRDSPD and FLSPD. Finally they are checked to ensure that their values have not decreased too far.

Dead

This routine prints the dead frog and decreases the value of LIVES.

The first step is to check whether the variable DED has been set by the collision routines which we will add next issue.

Lines 10820 to 10850 set up the sprite definitions to point to the correct sprites for a dead frog. 10860 to 11010 simply set up the correct X and Y co-ordinates of the frog (he would look pretty silly hovering in mid air!). Next the colours are set up in lines 11020 to 11040.

Now the interesting bit! Remember the interrupt handling routine, 'HANDLE' that we added in the second article? Well, unless we stop this from being called, the dead frog will continue moving (actually the background would move, not the frog).

To stop this from happening, we point the interrupt vector at $0314 and $0315 to the end of the routine, thus executing no code.

The rest of this section is simple to follow, and does this: Decrease Lives, pause, reprint status panel, pause, set up interrupts, refill FOOD, re-set Bird and Fly positions, re-set Cycle position.

The last pieces of code from line 11500 onwards, are simply delay loops used by the previous routine.

Next issue – collisions!

LANGUAGE LAB-C

This month David Janda begins a look at C — a very versatile programming language.

C IS AN APPLICATIONS LANGUAGE. That is, it's used by programmers to write a variety of programs such as text editors, programming utilities and such like. It is also the 'flavour of the day' in the micro world. Partly because of the popularity of the UNIX operating system (of which C is the language) and partly because of other features such as its compactness, portability and speed of execution.

The good thing about C is that it is a language for programmers. Many programming languages get designed by committee, but not C. C has gained its popularity not because of any hype or backing from a government (as the USA has backed ADA), but because programmers LIKE C.

So what is C, what can you do with it, and how good is it?

Why C?

C has many virtues. It is a modern language which incorporates modern control features. It is also a compact language; C can be installed on small micros – such as the 64 – and the code that it produces is compact and fast. Another benefit of C is that it is portable. We have all heard claim that this or that language is portable, only to discover that a major re-write is necessary before a program can be run on a different system. But with C it really is portable between different computers. If any changes to the source code are necessary, it usually means altering a few lines in a 'header' file which can accompany the main program.

For those of you who are interested in the 'real' world of programming, it is worth noting that C is the programming language of the UNIX operating system. UNIX is already an accepted operating system in the PC world, and the EEC have decided to adopt it as their official operating system.

C is a compiled language. The program (referred to as source code) is first written using a text editor or word processor. The source code is then submitted to the compiler, which, providing their are no errors, will compile the program into machine code and store it as a file which is called the object code.

The steps in developing a C program are a bit more complex than what was mentioned above, but it should give you a rough idea.

C BASICS

Here is a small C program:

```
1) # include <stdio.h>
2) main () /* This is a demo */
3) {
4) int num;
5) num = 1;
6) printf ("I am a simple");
7) printf ("computer. \ n");
8) printf ("My favourite number is %d
because it is first. \ n",num);
9)
```

This will print to the screen:

I am a simple computer.
My favourite number is 1 because it is first.

On a line analysis, this is what the program does (note line numbers are not part of C). Line one tells the compiler to incorporate information found in the file stdio.h. Every C program must incorporate at least one function, and line two identifies the function called main (the parentheses are there to identify main as a function). The open curly bracket identifies the beginning of the function body. Line four is a declaration statement, here the variable num has been declared of the type integer. In line five num has been assigned with the value one and lines six to eight print the output.

Line nine indicates the end of the function.

The '%d' instructs the micro where and in what form the value held in num is going to be displayed. The \ n means new-line carriage-return.

In next month's article I shall explain data types and control structures.

C Power V2.4 by Pro-Line Software Ltd

To the best of my knowledge, the C power package is the only C compiler available in the UK so far. By the time you read this a cheaper package called Super C should be available from First Software. The reason for this longer-than-normal review is to give you an idea of what the package offers and is capable of, as it is not the cheapest available!

On with the review! C Power is a complete C development package that will enable the user to produce stand-alone C programs. Most language packages available for the 64 fall into one of two categories: educational or development. Educational packages (such as the ADA tutorial) do not offer a comprehensive package that can be used to develop software. Instead, the emphasis is on introducing the user to the language. Development packages on the other hand (such as OCS and Orpheus Pascal) offer an excellent package, but are not too good when it comes to documentation. I was therefore surprised with C Power, because it is a very 'full' compiler offering lots of facilities as well as providing an excellent tutorial book that used in conjunction with the compiler will enable beginners to learn the language.

The package itself consists of a double sided disk which contains the compiler, shell, libraries and example programs. A 44 page user guide gives details on the compiler and very little else. The big plus is the C Primer Plus book that is also supplied with the package, and which can be regarded as a complete tutorial to C.

Implementation Details

One of the benefits of C is its portability – the ability to run C programs that were written on other machines. Because of this compatability is an important factor when developing a C package. Pro-Line

has got a very compatible package with C Power, but of course, there are some differences.

In brief, the omissions from standard C are: no bit field manipulation; static pointers may not be initialised except for character points initialised with strings; certain operators under certain conditions will not work unless the expressions are parenthesised. Most of these omissions are not serious, and can be got round quite easily.

The following table lists the size, in bytes of all data types supported by the compiler:

Type	Size
char	1
short	2
int	2
long	2
unsigned	2
float	5
double	5
pointer	2

Looking at the table, you can see that types short, int and long are the same, as are float and double. This practice is not uncommon in micro implementations of C, but is a strange one. For a package of this price I would have expected to see long and double supported.

The library supplied with the C Power compiler is quite standard. However, it would have been a good idea to include functions dependent on the 64, such as some sound and graphics functions.

Documentation

The user manual is supplied as sheets of paper which is stapled near the top. Punch holes enable the manual to be fitted into a ring binder, which is a good suggestion as my copy started to fall to bits after a short while. The manual itself makes no attempt to teach or introduce the user to C – that's left to the tutorial book. Instead the manual provides information on the implementation of C Power and descriptions of the editor, compiler, linker and so on. One section lists the functions provided with the library that is part of the package. The majority of functions are listed with name, number, order and types the function takes, description of the function and an example. A good idea this, as it will enable the user who wishes to port C source from another machine, to check up on functions to see if they are compatible.

The book 'C Primer Plus' by Waite, Prata & Martin (Sams, $19.95, ISBN 0-672-22090-3) is supplied with the C Power package. Quite simply it is the best language tutorial book I have ever read! This hefty tome (531 pages) takes the reader from the concept of programming right up to detailed discussion on C I/O. What's more, it is well written in a friendly (and amusing) manner with plenty of illustrations, summary pages and so on. Even if you don't intend to get the C Power package get this book!

C Power in Use

The user manual suggests that the system disk should be backed up. To do this, the shell, editor, syntax checker, linker and so on should be copied on to one disk. Another disk should be used to copy the flip side of the system disk; this contains the Stdlib.1 and Syslib.1 function libraries.

TABLE 1 — COMMANDS SUPPORTED BY THE SHELL

l	list work disk directory
ls	list system disk directory
rm	scratch file from work disk
mv	rename file on work disk
pr	list contents of a file on work disk
disk	send command string to work disk
load	load, but not run, command from work or system disk
work	show or set device and drive numbers
sys	as above but for system disk
ed	load and run editor
ced	load and run syntax checker
cc	compile C source
link	run linker

NOTE: All the commands are followed by arguments such as file names, drive or device numbers.

The compiler itself is copy protected (not very well though) so once working copies are made, three disks should be at hand: One containing the shell etc; another the libraries; and the third being the master disk.

The SHELL is the first program that is run when using C Power. SHELL itself is a mini-command interpreter. It supports command line arguments and I/O redirection along with the compiler and other programs that are designed to work under it. The nearest comparison would be Basic's screen editor, which can be used to develop, edit and run programs as well as issue I/O commands (such as disk directory, opening files etc). The command available from the Shell are listed in Table 1.

Entering some C source code first requires the editor to be loaded and run. This is simple done by entering 'ed' (optionally followed by a file name if an existing file is to be amended). The editor is a very comprehensive bit of code that provides numerous commands for moving about the text buffer. Once the code is written it can then be saved to disk. A syntax checker is also provided which does what its name suggests – checks the syntax of a C source file. The program is listed as it is being checked, and will stop if an error in syntax is found. If this is the case, a couple of key presses and you are back into the editor at the place where the error was found. Needless to say that the editor, syntax checker and all the other Shell commands are written in C!

The next stage is to invoke the compiler by entering cc filename. This loads and runs the compiler which produces an object code file.

The first stage is to load and run the linker. This will produce a runable file and three options are available. First it is possible to produce a C program that will run under the Shell (just like the Shell commands). It is also possible to specify a starting address, this means that the C program will have to be loaded and a SYS call made to the starting address. The third option is to produce a file that starts at the start of Basic.

Summary

Without a doubt C Power is a very powerful package. It is quite feasible that it can be used to develop commercial programs, and its numerous features give the user a great deal of flexibility.

The only drawback with C Power is its price. It's a very expensive package, and I would have expected such things as more C64 dependent functions, long integers and double precision floating point.

Having said that, its pros far outweigh the cons, and I would recommend the C Power package to the novice as well as the professional.

BETTER MATRIX

William Fong adds a little more power to your MPS

Simply type it in as you would any other program and SAVE it. When you RUN it, any typing errors in the data will be found and the line of the error given. Correct any offending lines and run again until the program runs without any errors. Do not attempt to use the program before all errors have been found or you could cause your machine to crash.

Once the program is in memory you can get rid of the loader by typing NEW. Then type SYS 49152 to initialise better matrix. A 3D title/menu page should be displayed showing you the simple controls that are needed to use the program.

Trying It Out

When everything is working type in the short program in Figure 2. Now press function key 3, which will give a blue border, and RUN the program. Not very nice is it?

Now press function key 1, which will give a cyan border, and RUN the program again. This time the letters will have descenders.

Better matrix should therefore give a better appearance to many of your printouts. However, as each word-processor is different it is impossible to say whether it will work with them loaded into your machine. At the moment better matrix sits at memory location $C000 (49152) so it will definitely not work with progrms that use this area of memory.

THE COMMODORE MPS 801 is an extremely popular printer because of its cheap price. If you are only after the occasional computer listing then it is quite adequate. A problem arises however if you wish to use the MPS 801 to produce quality text output as none of the lower case letters have descenders. This means that a letter 'J' would simply sit on the same line as a letter 'a'. This makes it very difficult to read large amounts of text.

If you want to enter the wonderul world of word-processing and are thinking of trading in your MPS 801 and purchasing a more expensive printer then wait a minute. Before you take any money out of your depleted bank account take a look at Better Matrix.

So how does it help? Well Better Matrix uses a similar method of printing as the more expensive near letter quality (NLQ) printers. First one part of the letter is printed then the printer goes back over the letter and adds the missing parts of the letter to the page.

By implementing this on the MPS 801 we can obtain an acceptable print resolution of 14 dots by 16 dots. Obviously this does not improve the definition of the character, as proper NLQ does, but it does allow you to print characters with descenders. Take a look at Figure 1 which shows the normal printout of the MPS 801 compared to that of the newer matrix.

However, printing in this manner does have a disadvantage. As the printer has to go over each line of text more than once it takes a lot longer to print out any text, this is not only common to the MPS 801 but all NLQ printer suffer in the same way. On the MPS 801 the speed is reduced from 50 characters per second to 20 characters per second.

Getting It In

Better matrix is quite short and the program is all in machine code but is presented here in the form of a Basic loader.

PROGRAM: BETTER MATRIX

```
2000 FOR L=0 TO 98:CX=0
   :FOR D=0 TO 15:READ A
   :CX=CX+A:POKE 49152+L*16+D,
   A:NEXT D
2010 READ A:IF A<>CX THEN PRI
   NT"ERROR IN LINE";
   2040+(L*10):STOP
2020 NEXT L:GOTO 4000
2040 DATA 120,169,192,141,21,
   3,169,73,141,20,3,88,169,6,
   141,33,1489
2050 DATA 208,141,32,208,162,
   0,189,246,194,32,210,255,
   232,224,255,208,2796
2060 DATA 245,162,0,189,245,
   195,32,210,255,232,224,255,
   208,245,162,0,2859
2070 DATA 189,244,196,32,210,
   255,232,224,255,208,245,
   162,0,189,243,197,3081
2080 DATA 32,210,255,232,224,
   43,208,245,96,165,203,201,
   4,208,18,169,2513
2090 DATA 119,141,38,3,169,
   192,141,39,3,169,3,141,32,
   208,76,49,1523
2100 DATA 234,201,5,208,15,
   169,202,141,38,3,169,241,
   141,39,3,169,1978
2110 DATA 6,141,32,208,76,49,
   234,72,165,154,201,3,240,5,
   144,3,1733
2120 DATA 76,135,192,104,76,
   202,241,104,141,100,194,72,
   138,72,152,72,2071
2130 DATA 173,100,194,201,80,
   208,3,76,6,193,201,81,208,
   3,76,32,1835
2140 DATA 193,201,71,208,3,76,
   58,193,201,74,208,3,76,84,
   193,201,2043
2150 DATA 89,208,3,76,110,193,
   201,70,208,3,76,136,193,
   201,13,208,1988
2160 DATA 3,76,36,194,201,65,
   208,3,76,162,193,201,67,
   208,3,76,1772
2170 DATA 188,193,201,69,208,
   3,76,214,193,201,82,208,3,
   76,240,193,2348
2180 DATA 201,87,208,3,76,10,
   194,201,32,144,19,201,145,
   240,15,201,1977
2190 DATA 146,240,11,174,99,
   194,169,32,157,245,194,238,
   99,194,104,168,2464
2200 DATA 104,170,104,76,221,
   237,160,7,185,101,194,32,
   221,237,185,109,2343
2210 DATA 194,174,99,194,157,
   245,194,238,99,194,136,16,
   235,76,93,194,2538
2220 DATA 160,7,185,117,194,
   32,221,237,185,125,194,174,
   99,194,157,245,2526
2230 DATA 194,238,99,194,136,
   16,235,76,93,194,160,7,185,
   133,194,32,2186
2240 DATA 221,237,185,141,194,
   174,99,194,157,245,194,238,
   99,194,136,16,2724
2250 DATA 235,76,93,194,160,7,
   185,149,194,32,221,237,185,
   157,194,174,2493
2260 DATA 99,194,157,245,194,
```

```
238,99,194,136,16,235,76,
93,194,160,7,2337
2270 DATA 185,165,194,32,221,
237,185,173,194,174,99,194,
157,245,194,238,2887
2280 DATA 99,194,136,16,235,
76,93,194,160,7,185,181,
194,32,221,237,2260
2290 DATA 185,189,194,174,99,
194,157,245,194,238,99,194,
136,16,235,76,2625
2300 DATA 93,194,160,7,185,
197,194,32,221,237,185,237,
194,174,99,194,2603
2310 DATA 157,245,194,238,99,
194,136,16,235,76,93,194,
160,7,185,205,2434
2320 DATA 194,32,221,237,185,
237,194,174,99,194,157,245,
194,238,99,194,2894
2330 DATA 136,16,235,76,93,
194,160,7,185,213,194,32,
221,237,185,237,2421
2340 DATA 194,174,99,194,157,
245,194,238,99,194,136,16,
235,76,93,194,2538
2350 DATA 160,7,185,221,194,
32,221,237,185,237,194,174,
99,194,157,245,2742
2360 DATA 194,238,99,194,136,
16,235,76,93,194,160,7,185,
229,194,32,2282
2370 DATA 221,237,185,237,194,
174,99,194,157,245,194,238,
99,194,136,16,2820
2380 DATA 235,76,93,194,169,8,
32,221,237,169,13,32,221,
237,169,15,2121
2390 DATA 32,221,237,173,99,
194,240,17,206,99,194,162,
255,232,189,245,2795
2400 DATA 194,32,221,237,236,
99,194,208,244,169,8,32,
221,237,169,13,2514
2410 DATA 32,221,237,169,15,
32,221,237,169,0,141,99,
194,104,168,104,2143
2420 DATA 170,104,96,0,0,15,
128,184,196,196,168,252,8,
15,128,128,1788
2430 DATA 128,128,128,143,8,
15,128,132,248,196,196,184,
8,15,128,132,1917
2440 DATA 143,128,128,128,8,
15,128,248,164,196,196,184,
8,15,128,135,1952
2450 DATA 136,136,136,132,8,
15,128,132,253,128,128,128,
8,15,128,128,1739
2460 DATA 135,136,136,132,8,
15,128,252,192,192,188,128,
8,15,128,135,1928
2470 DATA 136,136,132,128,8,
15,128,136,196,196,248,192,
8,15,128,128,1930
2480 DATA 128,128,143,128,8,
15,128,192,188,196,196,184,
8,15,128,200,1985
2490 DATA 196,196,196,184,8,
15,128,200,212,212,212,184,
8,15,128,136,2230
2500 DATA 132,132,248,132,8,
15,128,188,192,184,192,188,
8,15,128,128,2018
2510 DATA 128,128,128,128,8,0,
147,152,32,18,166,166,166,
166,166,166,1865
2520 DATA 166,166,166,166,166,
166,166,166,166,166,166,
166,166,166,166,166,2656
2530 DATA 166,166,166,166,166,
166,166,166,166,166,166,
166,166,166,166,13,2503
2540 DATA 159,18,176,195,195,
195,195,195,195,195,195,
195,195,195,195,195,2888
2550 DATA 195,195,195,195,195,
195,195,195,195,195,195,
195,195,174,152,166,3027
2560 DATA 166,166,166,166,166,
166,166,166,166,13,159,18,
194,32,66,69,2045
2570 DATA 84,84,69,82,32,77,
65,84,82,73,88,32,79,78,32,
77,1118
2580 DATA 80,83,45,56,48,49,
32,194,144,166,152,166,166,
166,166,166,1879
2590 DATA 166,166,166,166,13,
159,18,194,163,163,163,163,
163,163,163,163,2352
2600 DATA 163,163,163,163,163,
163,163,163,163,163,163,
163,163,163,163,163,2608
2610 DATA 163,163,194,144,166,
152,166,166,166,166,166,
166,166,166,166,13,2489
2620 DATA 159,18,194,32,87,82,
73,84,84,69,78,32,66,89,32,
32,1211
2630 DATA 87,73,76,76,73,65,
77,32,70,79,78,71,32,194,
144,166,1393
2640 DATA 152,166,166,166,166,
166,166,166,166,166,13,159,
18,194,32,32,2094
2650 DATA 32,32,32,32,32,32,
32,32,32,32,32,32,32,32,32,
32,512
2660 DATA 32,32,32,32,32,32,
32,32,194,144,166,152,166,
166,166,166,1576
2670 DATA 166,166,166,166,166,
13,159,18,194,32,40,67,41,
32,49,57,1532
2680 DATA 56,54,32,158,176,
195,195,195,195,195,195,
195,195,195,195,195,2621
2690 DATA 195,195,195,195,195,
195,195,195,195,195,174,
152,32,32,32,166,2538
2700 DATA 13,18,159,173,195,
195,195,195,195,195,195,
195,195,195,158,194,2665
2710 DATA 32,73,78,83,84,82,
85,67,84,73,79,78,83,32,32,
32,1077
2720 DATA 32,32,32,32,32,194,
152,32,166,166,166,13,32,
144,18,166,1409
2730 DATA 32,166,166,166,32,
166,32,166,32,158,194,163,
163,163,163,163,2125
2740 DATA 163,163,163,163,163,
163,163,163,163,32,32,32,
32,32,32,32,1691
2750 DATA 194,144,32,32,152,
32,166,13,32,152,18,166,32,
166,166,166,1663
2760 DATA 32,166,32,166,32,
158,194,32,70,85,78,67,84,
73,79,78,1426
2770 DATA 32,75,69,89,32,49,
32,58,32,79,78,32,194,144,
32,166,1193
2780 DATA 152,166,166,13,32,
152,18,166,32,32,32,166,32,
32,32,166,1389
2790 DATA 32,158,194,32,32,32,
32,32,32,32,32,32,32,32,32,
32,800
2800 DATA 32,32,32,32,32,32,
32,32,194,144,32,32,152,32,
166,13,1021
2810 DATA 32,152,18,166,166,
166,166,166,166,166,166,
166,166,158,194,32,2246
2820 DATA 70,85,78,67,84,73,
79,78,32,75,69,89,32,51,32,
58,1052
2830 DATA 32,79,70,70,194,144,
166,166,152,166,166,13,32,
152,18,166,1786
2840 DATA 32,32,32,166,32,32,
32,166,32,158,173,195,195,
195,195,195,1862
2850 DATA 195,195,195,195,195,
195,195,195,195,195,195,
195,195,195,195,195,3120
2860 DATA 189,144,32,166,152,
166,166,13,32,18,166,32,
166,32,166,32,1672
2870 DATA 166,32,166,32,166,
166,144,32,166,32,166,166,
166,166,166,166,2098
2880 DATA 166,32,166,166,32,
166,32,166,166,32,166,166,
32,166,152,166,1972
2890 DATA 166,13,32,18,166,32,
166,166,32,166,166,32,166,
32,32,32,1417
2900 DATA 144,32,166,32,32,32,
32,166,32,32,166,166,32,32,
166,166,1428
2910 DATA 32,166,166,32,166,
166,32,166,152,166,166,13,
32,18,166,32,1671
2920 DATA 166,166,32,166,166,
32,166,32,166,166,166,166,
166,166,166,32,2120
2930 DATA 166,166,166,166,32,
166,166,32,166,32,166,166,
32,166,166,32,1986
2940 DATA 166,166,166,13,32,
18,166,32,166,166,166,166,
166,32,166,32,1819
2950 DATA 166,166,166,166,32,
32,32,32,166,166,166,166,
166,32,32,166,1852
2960 DATA 166,166,32,32,166,
166,32,32,32,166,166,13,32,
18,166,166,1551
2970 DATA 166,166,166,166,166,
166,166,166,166,166,166,
166,166,166,166,166,2656
2980 DATA 166,166,166,166,166,
166,166,166,166,166,166,
166,166,166,166,166,2656
2990 DATA 166,166,166,13,32,
18,166,166,166,166,166,166,
166,166,166,166,2221
3000 DATA 166,166,166,166,166,
166,166,166,166,166,166,
166,166,166,166,166,2656
3010 DATA 166,166,166,166,166,
166,166,166,166,166,166,17,
17,158,13,0,2031
3020 DATA 253,254,253,254,253,
254,253,254,253,254,253,
254,253,254,253,254,4056
4000 PRINT "[CLEAR,DOWN3,
RIGHT6]BETTER MATRIX NOW
IN MEMORY"
4010 PRINT "[DOWN2,SPC6]TYPE
SYS 49152 TO ACTIVATE."
```

If you've never heard of bulletin boards or you just want to know how to get on one — read on. David Janda shows you how.

I HAVE BEEN ACCUSED OF DOMINATing Communication Corner with M'Net/CNET blurb. Well, as promised, here is (a refreshing) change. This month the subject is bulletin boards.

A bulletin board (BB from now on) can best be compared to a community bulletin board. A BB is usually run by a hobbiest in his/her own spare time on a home micro with disk drive and auto-answer modem. A BB will allow one user at a time to browse through the files stored on the board. These can include messages from other users, general information, specialists info, you name it you can put in on a BB.

Compared to Micronet or Compunet (I knew I would mention them sometime!) BBs are not technically brilliant, but they are by no means crude. One of the best things about using different BB's (board-walking) is that they are not bureaucratic, official or run by money makers – and that makes a difference!

What Type?

The UK currently has over 200 BBs that operate at regular times, and the number is growing. There are basically two types of BB from which to choose. First there is the traditional scrolling type of BB. To access this, you will need terminal emulation (often referred to as dumb terminal) software. Most comms packs include this type of facility. In case you are a Viewdata (Prestel) user who does not have this type of package, Dialsoft on page 600617160a at £1.50 should get you going. Compunet users can purchase TTY which is at 114007 for £4.99.

This type of software does not give you any colour or graphics, but enables you to log on to most types of BB.

The second type of BB operates on Viewdata (Prestel-like) standards. This type of board presents information with colour and low-res graphics in a page format. Micronet-Prestel subscribers will not need to buy any additional software, but Compunet subscribers will need the free Viewdata proggy at 2020. Again, as with the dumb terminal software, most comms software has a Viewdata mode.

Which Modem?

Most of the BBs are run at 300 baud. This means that you'll need a modem such as the Voyager 7, Nightingale or Multi modem from Miracle Technology. All three modems will also allow you to access boards at 1200/75 Prestel and scrolling formats. CBM modem owners need not feel hard done by. Many boards now allow access at 1200/75 baud, and some even operate at 1200/1200.

What's There?

Each BB has its own unique character. But most have an E-Mail (Electronic Mail) option that enables you to send and receive messages. These messages can be private, i.e. to another user, or be posted on the general board for everyone to see. Other features on BBs typically include free downloadable software. Downloading is done in several ways, but by far the most popular is the X-Modem format, so check to see if your comms package has this option. Another common feature to be found on BBs are the SIGs – Special Interest Groups. These are areas which contain information on one particular subject such as a micro, comms, politics or whatever.

There are many other features to be found on BBs. Some even have on-line adventures that you can play.

When using a BB, it's worth remembering that the service which you are using is two-way. That is, it's up to you, the user, to help supply the board with information. Have you got any software (of your own)? Then why not upload it for everyone to use. How about asking the System Operator (SYSOP) to set up an SIG dedicated to Commodore machines if there is not one there already?

The List

Opposite is a very small selection of BBs that are currently in operation. All the boards listed operate on a 24 hour basis. V/Data means that you will need Prestel type software to access the board. 1200-75 means that you need scrolling type software and a modem, such as the one from CBM, to access the service. Finally, 300 means 300-300 baud access which needs scrolling software. CBM modem users will NOT be able to access this type of board.

All the boards listed have a section which contains phone numbers for other boards. Have fun!

What About the 64?

To the best of my knowledge, there are only three bulletin boards which are run on the Commodore 64. One is in Aberdeen, one in Dublin and the third in Denmark! No doubt there are quite a few in the good ole' U$ of Reagan which I shall check out next month. I also understand that there is some public domain (free) BB software knocking about in Ireland which I shall also track down. As you can see below, running a Bulletin Board on a C-64 is perfectly feasible.

COMMUNICATION

CORNER

Name	Telephone	Rates
BABS 1	0394 276306	300
Brixton ITec	01 735 6153	V/Data
Cardif ITec	0222 464 725	V/Data
C-View	0702 546373	V/Data
FBBS Swansea	0792 203953	300
Gnome at Home	01 888 8894	V/Data
Hackney BB	01 985 3322	V/Data
Haunting Thunder	0752 364059	300
Livingston BBs	050 638526	300
London BB	01 455 6607	300/12-75
London Underground	01 863 0198	300/1200-75
Mactel	01 341 1719	300
Mailbox-80 (liverpool)	051 428 8924	300/1200-75
Metrotel	01 941 4285	V/Data
Norview	0604 20441	V/Data
TBBS Blandford	0258 54494	300
TBBS London	01 348 9400	300/1200-75/1200-1200

Below is a printout of part of a session on SBBS Aberdeen which is on 24 hrs a day, at 300 baud on 0224 781919, eight bits no parity. SBBS is run on a C128 with a 1541 disk drive. The 24K of software was written by Noel Gaddo, the Sysop using the Blitz Basic compiler, not bad eh!

The time is 04:33:15 One moment loading.
SPACE BAR=pause/resume, CTRL X=quit
SBBS Main Menu (1)
B-News/Bulletin Board info
G-Goodbye/Log-off
H-Help with this section
M-Messaging Area
U-User log
Y-Yell for SYSOP

2-Goto Main Menu (2)

?-Print this menu again

B,G,H,M,U,Y,2
or ? for menu :m

The time is 04:33:35 One moment loading.
SPACE BAR=pause/resume, CTRL X=quit

SBBS Messaging menu

G-General messages
H-Help with this section
P-Pen-pal messages
S-Sysop messages to/from

1-Goto Main Menu (1)
2-Goto Main Menu (2)
?-Print this menu again

G,H,P,S,Y,1,2
or ? for menu :g

The time is 04:33:55 One moment loading.
SPACE BAR=pause/resume, CTRL X=quit

SBBS General messages
C-Check Mail
D-Delete a message from file
H-Help with this section
M-Message catagories
Q-Quick scan of messages
R-Read message(s)
S-Send message
1-Main Menu (1)
2-Main Menu (2)

C,D,H,M,Q,R,S,Y,1,2
or ? for menu :r
One moment loading

First message £ 1 Last messge £ 34

Messge number: [From]-[To] <CR>=Exit
?:34-34

One moment loading.
Message 34 [general] section.

MESSAGE TO :ALL
MESSAGE FROM :FERGUS MCDONALD
SUBJECT :ANOTHER CBM BBS!!
DATE :THURSDAY 20/ 2/86
USER IS A :NON MEMBER.

—

Hi folks! I have set up a BBS in Dublin, Ireland, running on a Commodore 64 with WS2000 and 2 disk drives. I am a heavy CBM fan. The BBS software (by me) is called SIBBS. It is an interpreter – no menus! And it is all in machine code. Give it a ring and you will see. It is QUITE different. It is refreshing to see another BBS running on a Commodore 64. Also, what does All think of the new SFD 1001 CBM disk drive with 1MB per disk?? Anyway, the board (run by moi for the Irish Amateur Computer Club) is called the IACCBBS, and ring Dublin 903341 (24 Hours) on 8 bits, no par, 1 stop, 300 Baud

C,D,H,M,Q,R,S,Y,1,2
or ? for menu :1
One moment loading

Last but not Least

Well that's it for another month. I shall be reviewing at least two modems and some comms packages. Finally, a special mention to Richard James (CNET RBJ2) who wanted his name to appear in THE magazine in THE column. Nuff said! Don't forget to drop me a line on Compunet ID D.JANDA or Prestel 919992677.

CORNER

STATESIDE NEWS

Lewis Tilley gives you the update from across the Atlantic.

HAS THIS BEEN FOR COMMODORE the "winter of discontent", or just a winter of the big freeze? In the US programs like Freeze Frame from Cardco, and Isepic from Starpoint Software, seem to reflect in their names the low point reached by Commodore's stock. By midwinter it had dived to its lowest level ever of six, down from an all time high of 50 on the NY Stock Exchange.

Is this great company on its way out? How can a company that has sold over four million units of the C64 be in such trouble? Even this past Christmas it is rumoured that 60% of the sales were of the old reliable, that the C128 was so sluggish that an artificial shortage in the C64 was created by holding back supplies of them in the east coast. Other guesses were that the C64 was "warehoused" in order to raise its price and/or to re-introduce it in a new case with some fancy new touches. The above rumours, incidently, are through the courtesy of Felix Riveria who writes in what is perhaps the outstanding computer support group publication in the US. "The New York Commodore Exchange Network News" is published monthly in newspaper format. To subscribe, write Brian A Glover, Editor, 420 Clinton Avenue 5F, Brooklyn, NY 11238 USA. Its cost in the US $12.00 a year. Cheap, cheap, cheap.

My contact with user groups was greatly expanded by attendance at the 1986 West Coast Commodore Association "Commodore Show II" this February in San Francisco. At least a dozen groups were represented. Leading the field was the grand daddy of them all, The Toronto Pet Users Group, followed by such US giants as the Oregon based US Commodore Users Group, P.O. Box 2310, Roseburg, Oregon 97470 USA, and the above mentioned NYC conglomerate. No less than nine smaller California groups were listed in the show catalogue.

What's a BIG West Coast computer show like? Jammed packed with people and programs! The people were a little older on the average than those I've seen attending the shows at Earls Court or the International Commodore Show in London last year. But they were definitely not like the three-piece suit, business types that I ran into in San Francisco at an IBM compatible software show that was being held at the same time.

Now for the programs which I mentioned earlier. Freeze Frame by CARDCO, Inc., 300 S Topeka, Wichita, KS 67202 will quietly sit inside your computer, totally transparent to any other programs you may wish to load and use. Then when you want to make a screen dump, you call it with two key strokes and voila!, the frame is frozen and sent to your printer. You then continue to run whatever you may have as your main program.

Don't confuse two different programs coming from the US both called Snapshot. One is a utility from COMPUTE!s Gazette magazine written in machine language which can store whatever you have on the TEXT screen (up to 53 screens) and then retrieve it for display or for printing. The other is a cartridge named Snapshot 64 which comes from those fine people at CSM Software, Inc., P.O. Box 563, Crown Point, Indiana 46307 who developed the best of the 1541 Disk Drive Alignment Programs (buy the Version 2.0 at $44.95 plus shipping).

Snapshot 64 is touted as the "ultimate" backup utility since you are able to "stop most any program after the protection check and then resume...totally bypassing the protection check." Price on this utility is $49.95 plus shipping costs.

Remember the exciting advertisements from Starpoint Software of Gazelle, CA 960347? They announced a utility called STARDOS which would do everything (even makes great coffee). Well, they were blocked from releasing it by a lawsuit slapped on them by SKYLES Electric Works, 231E South Whisman Road, Mountain View, CA94041. Skyles says that is was too much like their cartridge 1541 FLASH (£59.95 – on sale, plus $15.00 UK shipping charge) which was developed by the same computer expert.

Starpoint is delivering one of the bargains of the year. They are marketing a 256K RAM board for the Amiga functionally identical to the Commodore 1050 RAM board for £99.95 + $6.00 shipping outside the USA.

Isepic, which is also a Starpoint product at $64.95, is a combination hardware and software package that closes this little survey of new utilities. It "de-protects" by capturing and saving the protected program as it runs in the 64's memory. "This 'snapshot' [so help me, they advertise this one as a "snapshot", too] becomes accessible to the user for complete inspection and alteration." The quotation is from Starpoint's advertisement.

The midnight modem madness continues. Commodore associated QUANTUM LINK is giving away a 300 baud autodial modem if you subscribe for four months at $9.95 monthly. Viewtron charges you only $49.95 for a 300 baud Volks 6420 by Anchor Automation. They throw in a first hour on Viewtron for free. Playnet gets a bit tricky in its combination offer of a magazine subscription to AHOY, a Playnet membership kit and a modem if you subscribe to Playnet for three months and pay a first time membership fee of $19.95.

The thaw has certainly come to Amiga. Hardware prices are falling and software is flowing from the developers like warmed up maple syrup. The Electronic Arts Deluxe Paint program begins to really show what this amazing machine can do in presenting visual ideas. What's more it integrates with the Deluxe Graphics, Music and Printing modules of the same company. The games have arrived too. Flight Simulator from Sublogic, Napoleon at Waterloo by Krentek Software, Skyfox and a re-issue of the Software Golden Oldies, i.e. Pong, Eliza, Adventure and Life from EA are available now.

The C128 is less fortunate in the issuance of new programs designed especially for it. True, some very fine old programs are being updated such as Vizawrite and SuperScript. And now that Commodore has issued a new version of CP/M for the 128, which really works this time, all those wonderful old CP/M business programs are there for the taking.

I may be including a mention of the PC 10 in upcoming columns if the announcement by a PR (that's Public Relations in US english) man for X-PRESS Information Services is true. Commodore is going to try and hit the business world with a wire service which is compatible with IBM machines. They'll distribute the PC 10 in the US to clients who want the service compatible with IBM and may even offer a free Plus/4 to the less fluent businesses who want to try their service. This will all be tied in with QuantumLink to give a truly total service as well as find a use for all those great (and unsold) PLUS/4's.

Joe Nicholson continues his look at the C-16. This month — clocks and timers.

PROGRAMMING THE C16

IN THIS ARTICLE I SHALL ATtempt to explain some of the techniques involved in timing, interrupts and so on. I shall start with the keyboard interrupt as this is the easiest to explain. Every 1/50th of a second the computer interrupts its normal processing to execute a 'service routine'. This updates the clock and reads the keyboard, putting any new keys pressed into the keyboard buffer. It then resumes processing. The address contained in bytes $0314 and $0315 are the low and high bytes of the indirect values for the location of the interrupt. These values can be redirected in order to make the C-16 jump to your own machine code routine. This method was used in the play routine and the synthesiser article explained last month. After the user routine has been completed the program can then jump back into the service routine to update the counter etc. Alternatively it is possible to jump straight back into processing.

The interrupt is normally on. The machine code instruction SEI turns the interrupt off and CLI will turn it on again. The following routine will set the interrupt vector:

```
SEI        interrupt off
LDA low byte
STA $0314
LDA high byte
STA $0315
CLI         interrupt on
RTS         return
```

Similarly to restore the original interrupt:

```
SEI
LDA $0E
STA $0314
LDA $CE
STA $0315
CLI
RTS
```

Note that most of the programming associated with interrupts has to be done in machine code for speed, for instance it is not possible to disable the interrupts from Basic. At the end of the user interrupt routine use the instruction JMP $CE0E to return to the service routine. To jump back from the user routine to continue processing, ignoring the C-16's service routine, use the instruction JMP $FCB6 (IRQ exit). I won't include a demonstration of this type as the play command published in the December 1985 article on sound, and the sound synthesiser article published last month serve as fitting demonstrations.

Internal Timers

There are three internal 16 bit timers in the C-16. The timers operate at a frequency of 885 KHz on our PAL system machines. It therefore takes on 0.0741 seconds to count all the way from 65535 to zero. Timer #1 has the facility of being able to activate an interrupt upon reaching zero. Each timer is arranged as two eight bit registers in memory, using the normal protocol of high byte (i.e. multiples of 256) last, preceded by the low byte (remainder 0-255).

The registers are arranged in memory as follows:

$FF00 Timer #1 low byte
$FF01 Timer #1 high byte
$FF02 Timer #2 low byte
$FF03 Timer #2 high byte
$FF04 Timer #3 low byte
$FF05 Timer #3 high byte

To set a timer simply load the timer's registers with the starting value; it will then count down to zero. As the timer will obviously have to be set in two parts, (one POKE or STA for the low byte and one for the high byte) there should therefore be a delay of no greater than 125 µs between writing the low byte and the high byte, otherwise the timer will start to count down and therefore to be set incorrectly.

To eliminate this problem, therefore, the timer registers should be set in machine code using the following type of routine:

```
SEI disable interrupts. (we don't want a keyboard interrupt to occur between writing the low byte and the high byte)
LDA low byte of start time
STA Low byte of timer
LDA high byte of start time
STA high byte of timer
CLI turn the interrupts back on
RTS return
```

The Timers and Interrupts

Timer #1 is a more sophisticated timer than timers #2 and #3. When this timer is written to, it sets the timer to that value as expected, but also sets the 'timer #1 reload registers' to that value. The timer then counts down to zero, at which point an interrupt is generated if bit three of the interrupt mask flag is set. Bit three of the interrupt status register is then set. The timer is then reset to the reload value, and the counter carries on decrementing until it reaches zero when another interrupt is generated etc.

Timer #1 is potentially a very useful timer allowing periodic interrupts of specified delay length, for instance timers are used to generate accurate inter bit delays in a fast tape loading system which is under development.

Timer #2 and timer #3 are simpler timers. These timers go back to 65535 after they have reached zero, instead of being reset to a reload value. They still have the ability to generate interrupts when they reach zero however. To turn interrupt for timer #2 off/on, set/reset bit six of the interrupt mask register. When the interrupt takes place, bit six of the interrupt status register is set high.

The Interrupt Mask

The interrupts are turned on and off by setting/resetting bits on a register called the interrupt mask at $FF0A or decimal 65290. Bits one to seven control the following:

Bit 1 raster interrupt
Bit 2 light pen
Bit 3 timer 1 interrupt
Bit 4 timer 2 interrupt
Bit 6 timer 3 interrupt
Bit 7 interrupt request

Setting a bit high on the mask byte will enable the appropriate interrupt. Bit 2 light pen? There is no connection for a light pen on the edge connector, but there may be on the joystick port.

The Interrupt Status Register

This byte (at $FF09 or 65289 decimal) records which interrupt has interrupted. It is important for the interrupt service routine to know just which interrupt has been used so that it knows how to act. The arrangement of bits in this register is the same as the interrupt mask register (e.g. Bit four is the timer #3 interrupt bit). Strangely, to set a bit 'on' in this register, write that bit with a zero. Similarly write the bit with a one to reset that bit. Any interrupts from the C-16 are recorded by the C-16 setting the appropriate bit of this register.

Timer Interrupts

It is interesting to note that the three timer interrupts can still be used even when the raster and interrupt requests have been disabled with the SEI command. This is the technique used in the saving and loading of programs in the C-16. The routine at $E364, for instance (Figure 1) is used before loading/saving a block/header in the C-16. The routine at $E378, shown in Figure 2, is the opposite of the routine of $E364 and is used after loading/saving a block/header.

However it is not always necessary to perform timing exercises using the method

described above, thanks to a very useful ROM routine at $E2EA. If quite accurate time delays are all that are required, the routine at $E2EA can be used. Calling this routine will generate a delay of 1/50th second. This routine is used to generate delays in the saying/loading routine, e.g. the 3/5 second between printing SAVING or LOADING and turning the screen off on saving/loading is:

```
LDX # $1E
:L1 JSR $E2EA
DEX
BNE 4:L1
CCI
...
```

Note that as the routine at $E2EA turns the interrupts off to get better timing, the CCI command is used to re-enable them if you want them on again.

The Clock

The registers $0312 and $0313 are the low and high byte pointers for the 'update clock' routine which is called every 1/50th second by the C-16's service routine. This is usually set to $CE42 but can be re-directed to go to a user routine. Figure 3 shows the assembly text for a clock which displays its time continuously in the top right hand corner of the screen. It also has an alarm with sound and a facility to jump to a machine code routine upon the event of an alarm. The clock redirects the pointer $0312 and $0313 (786 and 787) to a new service routine stored at $0660. The whole machine code program is 308 bytes long and resides between $0600 and $0728, a free area of memory in the C-16. To type this in use the C-16 Assembler published in the June 1985 copy of Your Commodore. Alternatively Figure 4 shows the code for this program in data statements with a machine code loader at 10000 to POKE the bytes in. The time and the alarm time can all be set with Basic commands. The routine is stored at $0600 (1536 decimal).

Description of Clock Program

Lines 10120-10220 initialise the routines own variables.
Lines 10300-10380 turn off the clock by relocating the 'update clock' routine to its original value.
Lines 10500-10580 turn on clock; diverts 'update clock routine' to new service routine.
Lines 10600-10700 set time. To set the time, type: SYS1577,12:43:23. The latter three numbers are the time in hours, minutes and seconds. The system for scanning the line used in this subroutine will be explained in a later article on extending the Basic. Line 10690 then turns the clock on.
Lines 10800-10910 set the alarm. This works in the same way as the 'set time' routine. It is located at $103D (4157 decimal), so SYS1601,12:43:23 sets the alarm and initialises it so that it sounds for 30 seconds when the alarm time is reached.
Lines 11000-11990 – clock service routine.
Lines 11000-11990 count each 1/50th second jumping to $CE%2 if the next second has not been reached.
Lines 11200-11340 update seconds.
Lines 11400-11500 update minutes.
Lines 11550-11690 update hours, switching back to 00:00:00 after it reaches 23:59:59.
Line 11700 calls the routine which handles the alarm sound if it is on.
Line 11710 calls the r utine to print the time on the screen.
Lines 11800-11860 check to see if the alarm time has been reached by comparing the time ($D1-$DB) and the alarm time ($D9$F0) byte by byte.
Lines 11900-11990 the alarm has been reached. Line 11910 calls the alarm sound subroutine which starts the alarm sound if it is required.
Lines 11950-11980 call a machine code routine whose address is stored in bytes $E2 and $E3, if $E8 is set.
Lines 12000-12200 – Alarm routine. This handles the 30 second bleeping alarm if it is on.
Lines 12105-12200 are concerned with marking the bleeping.
Lines 12500-12680 initialise alarm sound. If an alarm has been reached and the register $E4 is > 0 to signify that the alarm sound is enabled, an alarm sound is generated in voice 2.
Lines 12550-12570 set the duration to 30 seconds.
Lines 12580-12610, set the frequency.
Lines 12620-12650 'on' voice 2.
Lines 12660-12670 select 29 bleeps.
Lines 13000-13150 Print time. This prints the time in the top right hand corner of the screen if $E9 is > 0.
Lines 13050-13090 set up the colour for the text. This is initially set to 16 (black) but can be set to other colours by POKEing address $E7.
Lines 13100-13150 print the characters of the time on the screen.

Figure 5 shows a list of all the system variables used in the program.

Note that the machine code jump routine cannot last for any longer than 1/50th second. At the end of the routine a JMP$CE42 should be used to exit the routine.

In conclusion, to use the clock, first of all initialise by typing in SYS1536.
To set the time: SYS1577, 12:23:36
To set the alarm: SYS1601, 07:35:26
To turn off the clock: SYS1551

PROGRAM : CLOCK CODE

```
10000 A=1536:DO:B=0:FORC=1TO
16:READD:IFD=-1THENPRINT"OK.
.":END
10010 B=B+D:POKEA,D:A=A+1:NE
XT:READD
10020 IFD<>BTHENPRINT"DATA E
RROR IN LINE";PEEK(63)+256*P
EEK(64):END:ELSELOOP
20000 DATA 169,1,133,225,169
,16,133,231,169,0,133,229,13
3,232,96,120, 2189
20010 DATA 169,66,141,18,3,1
69,206,141,19,3,88,96,120,16
9,96,141, 1645
20020 DATA 18,3,169,6,141,19
,3,88,96,162,0,32,115,4,149,
209, 1214
20030 DATA 232,138,201,8,208
,245,32,28,6,32,115,4,169,0,
133,208, 1759
20040 DATA 96,162,0,32,115,4
,149,217,232,138,201,8,208,2
45,169,1, 1977
20050 DATA 133,228,32,115,4,
96,255,0,255,0,255,0,255,0,2
55,0, 1883
20060 DATA 230,208,165,208,2
01,50,240,3,76,66,206,162,48
,160,58,169, 2250
20070 DATA 0,133,208,230,216
,196,216,208,56,134,216,230,
215,165,215,201, 2839
20080 DATA 54,208,46,134,215
,230,213,196,213,208,38,134,
213,230,212,165, 2709
20090 DATA 212,201,54,208,28
,134,212,230,210,201,52,208,
10,196,210,208, 2574
20100 DATA 16,134,210,230,20
9,208,10,165,209,201,50,208,
4,134,210,134, 2332
20110 DATA 209,32,207,6,32,1
7,7,162,8,181,208,213,216,20
8,13,202, 1921
20120 DATA 208,247,32,234,6,
165,232,240,3,108,226,0,76,6
6,206,165, 2214
20130 DATA 229,240,22,198,22
9,165,229,41,1,240,2,169,32,
133,230,173, 2333
20140 DATA 17,255,41,159,5,2
30,141,17,255,96,169,228,201
,0,240,0, 2054
20150 DATA 169,8,141,255,4,1
41,253,4,169,2,141,16,255,16
9,240,141, 2108
20160 DATA 15,255,173,17,255
,41,144,9,40,141,17,255,169,
29,133,229, 1922
20170 DATA 96,165,225,201,0,
240,20,162,8,165,231,157,31,
8,202,208, 2119
20180 DATA 250,162,8,181,208
,157,31,12,202,208,248,96,0,
0,0,0, 1763,-1
```

PROGRAM : CLOCK TEXT

```
10 GOTO900
100 E=0:B=0:I=0:E$=""
110 D$=MID$(A$,A,1):IFD$=B$O
RD$=""ORD$=" "THENRETURN
112 IFD$=":"ORE=2THENE$=E$+D
$:E=2:GOTO140
113 IFD$>"/"ANDD$<"6"THEN130
114 IFD$="("ORD$="*"THEN140
115 IFD$="$"THENI=1:GOTO140
120 E=1:RETURN
130 IFI=0THENB=(B*10)+VAL(D$
):ELSEB=(B*16)+DEC(D$)
140 A=A+1:GOTO110
200 READA$:IFA$="END"THENRET
```

▶

```
URN
210 IFA$=""ORASC(A$)=59THEN2
00
250 C$=LEFT$(A$,3):D$=RIGHT$
(A$,LEN(A$)-3+(LEN(A$)>3)):H
=PEEK(63)+256*PEEK(64):RETUR
N
400 A$=D$:B$=","
410 GOSUB100
420 IFE=1THENPRINT"BYT ERROR
 IN "H:RETURN
430 POKED,B:D=D+1:S=S+1:A=A+
1:IFD$=""THENRETURN:ELSEGOTO
410
500 K=1:A$=D$:IFA$=""THENRET
URN
510 Q$=LEFT$(A$,1):IFQ$="#"T
HENB$="":K=2:GOTO800
520 IFQ$="A"THENK=3:RETURN
530 IFRIGHT$(A$,2)="X)"THENB
$=",":K=11:GOTO800
540 IFRIGHT$(A$,1)="Y"ANDQ$=
"("THENB$=")":K=10:GOTO800
550 IFRIGHT$(A$,1)=")"ANDQ$=
"("THENB$=")":K=4:GOTO800
560 IFRIGHT$(A$,2)=",X"ORRIG
HT$(A$,2)=",Y"THENB$=",":K=5
+(RIGHT$(A$,1)="Y"):GOTO800
570 IFQ$="R"THENA=A+1:K=9:B$
="":ELSEK=12
800 GOSUB100
810 IFE<>2THENGOTO870
820 IFP=1THENB=0:GOTO870
850 FORI=0TOT:IFRIGHT$(E$,2)
=L$(I)THENB=L%(I):GOTO870
860 NEXTI:PRINT"LABEL NOT FO
UND IN"H:B=0
870 IF(K=5ORK=6ORK=12)AND(B>
255ORE=2)THENK=K+1-(K<12)
880 RETURN
900 CLR:DIML$(100):DIML%(100
):KEY2,"DATA"+CHR$(34):KEY3,
"LIST10000-"+CHR$(13)
910 KEY6,"DIRECTORY"+CHR$(13
)
1000 INPUTA$:IFA$="Q"THENEND
1010 A=1
2000 IFA$="P1"THENU=1:OPEN4,
4:CMD4:PRINT:GOTO1000
2040 IFA$="P0"THENU=0:PRINT#
4:CLOSE4:GOTO1000
2070 IFA$="LIST"ORA$="L"THEN
GOTO2700
2080 IFA$="SYMBOL"ORA$="S"TH
ENPRINT"SYMBOL TABLE":FORG=0
TOT:PRINTL$(G)"    AT:"L%(G)
:NEXT
2100 IFA$="ASSEMBLE"ORA$="A"
THEN5000:ELSE1000
2700 G=0:INPUT"START:";G:IFG
THENRESTOREG:ELSERESTORE9999
2750 READA$:IFA$="END"THENPR
INT">> OK.":GOTO1000
2760 PRINTPEEK(63)+256*PEEK(
64)" "A$
2770 IFU=0THENGETA$:IFA$<>""
THEN1000
2780 GOTO2750
5000 PRINT"(DOWN)ASSEMBLE":P
RINT"(DOWN)1ST PASS":T=-1:D=
0:S=0:P=1:RESTORE9999
5100 GOSUB200:IFA$="END"THEN
5500:ELSE7000
5110 GOSUB8000:RESTOREF:READ
A$:GOTO5100
5500 PRINT"(DOWN)2ND PASS":D
=0:S=0:P=2:RESTORE9999
5510 GOSUB200:IFA$="END"THEN
PRINT">> OK.":GOTO1000
5515 IFMID$(A$,4,1)="="THEN5
510
5520 IFASC(C$)=58THENC$=LEFT
$(D$,3):D$=RIGHT$(D$,LEN(D$)
-3+(LEN(D$)>3)):GOTO5520
5530 GOSUB8000:RESTOREF:READ
A$:GOTO5510
7000 IFASC(C$)<>58THEN5110
7010 T=T+1:IFT>100THENPRINT"
TOO MANY LABELS AT "H:GOTO10
00
7013 L$(T)=RIGHT$(C$,2)
7016 IFMID$(A$,4,1)="="THENB
$="":A=5:GOSUB800:L%(T)=B:GO
TO5100
7020 L%(T)=S:C$=LEFT$(D$,3):
D$=RIGHT$(D$,LEN(D$)-3+(LEN(
D$)>3)):GOTO5110
8000 A=1:B$="":F=H:IFC$="ORG
"THENA$=D$:GOSUB100:D=B:S=B:
RETURN
8010 IFASC(C$)=59THENRETURN
8020 IFC$="TRU"THENA$=D$:GOS
UB100:S=B:RETURN
8100 IF C$="TEX"THENFORB=1TO
LEN(D$):POKED,ASC(MID$(D$,B,
1)):S=S+1:D=D+1:NEXT:RETURN
8150 IFC$="BYT"THENGOTO400
9000 GOSUB500
9100 RESTORE9500+((ASC(LEFT$
(C$,1))-65)*10)
9110 READF$
9120 IFLEFT$(F$,2)=RIGHT$(C$
,2)THEN9200
9130 A=INSTR(F$,",",3):IFA<>
0THENF$=RIGHT$(F$,LEN(F$)-A)
:GOTO9120
9140 IFRIGHT$(F$,1)="←"THENP
RINT"SYNTAX ERROR IN "H:RETU
RN:ELSE9110
9200 FORI=3TOLEN(F$)STEP3
9210 E$=MID$(F$,I,1):IFE$<"0
"ORE$="←"THEN9240
9220 IFDEC(E$)=KTHEN9300
9230 NEXTI
9240 PRINT"ADDRESSING MODE E
RROR IN "H:RETURN
9300 G=DEC(MID$(F$,I+1,2)):P
OKED,G:D=D+1:S=S+1
9370 IFK=2ORK=5ORK=6ORK=11OR
K=10ORK=12THENPOKED,B:GOTO94
80
9380 IFK=7ORK=8ORK=13ORK=4TH
EN9450
9400 IFK<>9THENRETURN
9405 IFP=1ORB=0THENGOTO9480
9410 I=B-S-1:IFI<0ANDI>-129T
HENI=256+I
9420 IFI<0ORI>255THENPRINT"R
ELATIVE JUMP OUT OF RANGE IN
 "H:I=0
9430 POKED,I:GOTO9480
9450 POKED,B-(INT(B/256)*256
):POKED+1,INT(B/256):S=S+2:D
=D+2:RETURN
9480 D=D+1:S=S+1:RETURN
9500 DATA"DCD6DC6526977D879B
61A71575
9502 DATA"NDD2DC2522973D839B
21A31535
9504 DATA"SL30AD0EC0671E516←
9510 DATA"CC990,CS9B0,EQ9F0,
ITD2CC24,MI930,NE9D0,PL910,R
K100,VC950,VS970←
9520 DATA"LC118,LD1D8,LI158,
LV1B8,MPDCDCC52C97DD8D9BC1AD
15D5
9522 DATA"PXDECCE42E0,PYDCCC
C42C0←
9530 DATA"ECDCECC67DE5D6,EX1
CA,EY188←
9540 DATA"ORD4DC4524975D859B
41A51555←
9580 DATA"NCDEECE67FE5F6,NX1
E8,NY1C8←
9590 DATA"MPD4C46C,SRD20←
9610 DATA"DADADCA52A97BD8B9B
A1AB15B5
9612 DATA"DXDAECA62A28BE6B6,
DYDACCA42A07BC5B4,SR34AD4EC4
675E556←
9630 DATA"OP1EA
9640 DATA"RAD0DC0520971D819B
01A11515←
9650 DATA"HA148,HP108,LA168,
LP128←
9670 DATA"OL32AD2EC2673E536,
OR36AD6EC6677E576,TI140,TS16
0←
9680 DATA"BCDEDCE52E97FD8F9B
E1AF15F5,EC138,ED1F8,EI178
9682 DATA"TAD8DC8579D899B81A
91595,TXD8EC86696,TYD8CC8459
4←
9690 DATA"AX1AA,AY1A8,SX1BA,
XA18A,XS19A,YA198←
9999 REM
10000 DATA";C-16 CLOCK WITH
ALARM
10010 DATA";(C)1986 JOE NICH
OLSON
10100 DATA"ORG $0600
10110 DATA";
10120 DATA";INITIALISE
10150 DATA"LDA #1
10160 DATA"STA $E1
10170 DATA"LDA #16
10180 DATA"STA $E7
10190 DATA"LDA #0
10200 DATA"STA $E5
10210 DATA"STA $E8
10220 DATA"RTS
10300 DATA";
10310 DATA";TURN OFF CLOCK
10320 DATA"SEI
10330 DATA"LDA #$42
10340 DATA"STA $0312
10350 DATA"LDA #$CE
10360 DATA"STA $0313
10370 DATA"CLI
10380 DATA"RTS
10500 DATA";
10510 DATA";TURN ON CLOCK
10520 DATA":T5 SEI
10530 DATA"LDA #$60
10540 DATA"STA $0312
10550 DATA"LDA #$06
10560 DATA"STA $0313
10570 DATA"CLI
10580 DATA"RTS
10600 DATA";
10610 DATA";SET TIME
10620 DATA"LDX #0
10630 DATA":T1 JSR $0473
10640 DATA"STA $D1,X
10650 DATA"INX
10660 DATA"TXA
10670 DATA"CMP #8
10680 DATA"BNE R:T1
10690 DATA"JSR :T5
10695 DATA"JSR $0473
10697 DATA"LDA #0
10698 DATA"STA $D0
10700 DATA"RTS
10800 DATA";
10810 DATA";SET ALARM
10820 DATA"LDX #0
10830 DATA":T3 JSR $0473
10840 DATA"STA $D9,X
10850 DATA"INX
10860 DATA"TXA
```

```
10870 DATA"CMP #8
10880 DATA"BNE R:T3
10890 DATA"LDA #1
10900 DATA"STA $E4
10905 DATA"JSR $0473
10910 DATA"RTS
11000 DATA";
11010 DATA";CLOCK SERVICE RO
UTINE
11050 DATA"ORG $0660
11100 DATA";UPDATE JIFFY
11110 DATA"INC $D0
11120 DATA"LDA $D0
11130 DATA"CMP #50
11140 DATA"BEQ R:L1
11150 DATA"JMP $CE42
11200 DATA";UPDATE SECONDS
11210 DATA":L1 LDX #48
11220 DATA"LDY #58
11225 DATA"LDA #0
11230 DATA"STA $D0
11240 DATA"INC $D8
11250 DATA"CPY $D8
11260 DATA"BNE R:L2
11300 DATA"STX $D8
11310 DATA"INC $D7
11320 DATA"LDA $D7
11330 DATA"CMP #54
11340 DATA"BNE R:L2
11400 DATA";UPDATE MINUTES
11410 DATA"STX $D7
11420 DATA"INC $D5
11430 DATA"CPY $D5
11450 DATA"BNE R:L2
11460 DATA"STX $D5
11470 DATA"INC $D4
11480 DATA"LDA $D4
11490 DATA"CMP #54
11500 DATA"BNE R:L2
11550 DATA";UPDATE HOURS
11560 DATA"STX $D4
11570 DATA"INC $D2
11580 DATA"CMP #52
11590 DATA"BNE R:L3
11600 DATA"CPY $D2
11610 DATA"BNE R:L2
11620 DATA"STX $D2
11630 DATA"INC $D1
11640 DATA"BNE R:L2
11650 DATA":L3 LDA $D1
11660 DATA"CMP #50
11670 DATA"BNE R:L2
11680 DATA"STX $D2
11690 DATA"STX $D1
11700 DATA":L2 JSR :AS
11710 DATA"JSR :PT
11800 DATA";CHECK IF ALARM
11810 DATA"LDX #8
11820 DATA":L5 LDA $D0,X
11830 DATA"CMP $D8,X
11840 DATA"BNE R:LC
11850 DATA"DEX
11860 DATA"BNE R:L5
11900 DATA";ALARM!
11910 DATA"JSR :OS
11950 DATA"LDA $E8
11970 DATA"BEQ R:LC
11980 DATA"JMP ($E2)
11990 DATA":LC JMP $CE42
12000 DATA";
12010 DATA";ALARM SUBROUTINE
12020 DATA":AS LDA $E5
12030 DATA"BEQ R:L8
12100 DATA"DEC $E5
12105 DATA"LDA $E5
12110 DATA"AND #1
12130 DATA"BEQ R:L9
12140 DATA"LDA #$20
12150 DATA":L9 STA $E6
12160 DATA"LDA $FF11
12170 DATA"AND #$9F
12180 DATA"ORA $E6
12190 DATA"STA $FF11
12200 DATA":L8 RTS
12500 DATA";
12510 DATA";ON ALARM SOUND
12520 DATA":OS LDA #$E4
12530 DATA"CMP #0
12540 DATA"BEQ R:L7
12550 DATA":L7 LDA #F8
12560 DATA"STA $04FF
12570 DATA"STA $04FE
12580 DATA"LDA #$02
12590 DATA"STA $FF10
12600 DATA"LDA #$F0
12610 DATA"STA $FF0F
12620 DATA"LDA $FF11
12630 DATA"AND #$90
12640 DATA"ORA #$28
12650 DATA"STA $FF11
12660 DATA"LDA #29
12670 DATA"STA $E5
12680 DATA":L7 RTS
13000 DATA";
13010 DATA";PRINT TIME
13020 DATA":PT LDA $E1
13030 DATA"CMP #0
13040 DATA"BEQ R:LD
13050 DATA"LDX #8
13060 DATA"LDA $E7
13070 DATA":LA STA 2079,X
13080 DATA"DEX
13090 DATA"BNE R:LA
13100 DATA"LDX #8
13110 DATA":LB LDA $D0,X
13120 DATA"STA 3103,X
13130 DATA"DEX
13140 DATA"BNE R:LB
13150 DATA":LD RTS
63000 DATA"END
```

■

PROGRAMMING THE 1541

Bill Bremner begins a series designed to help you get the most out of your disk drive.

BEFORE I TRY TO EXPLAIN HOW THE 1541 Disk Drive works, let us discover what brought about its unique design and unravel some of its peculiarities.

Most modern computers employ an all-singing all-dancing (albeit expensive) Floppy Disk Controller or FDC to connect up to a disk drive. A software package called a Disk Filing System (DFS) or Disk Operating System (DOS) is then loaded into RAM or plugged in as ROM and manages the controller, to perform such tasks as formatting, reading, and writing. Of course, the more complex the DFS or DOS required, the more space is taken up inside the computer (the BBC micro uses the plug in ROM method, Apple and Atari both load their software into RAM). There are quite a few advantages in using this type of system: fast loading and saving, and a variety of increasingly complicated and fashionable drives. However, the restraint on memory usage restricts the

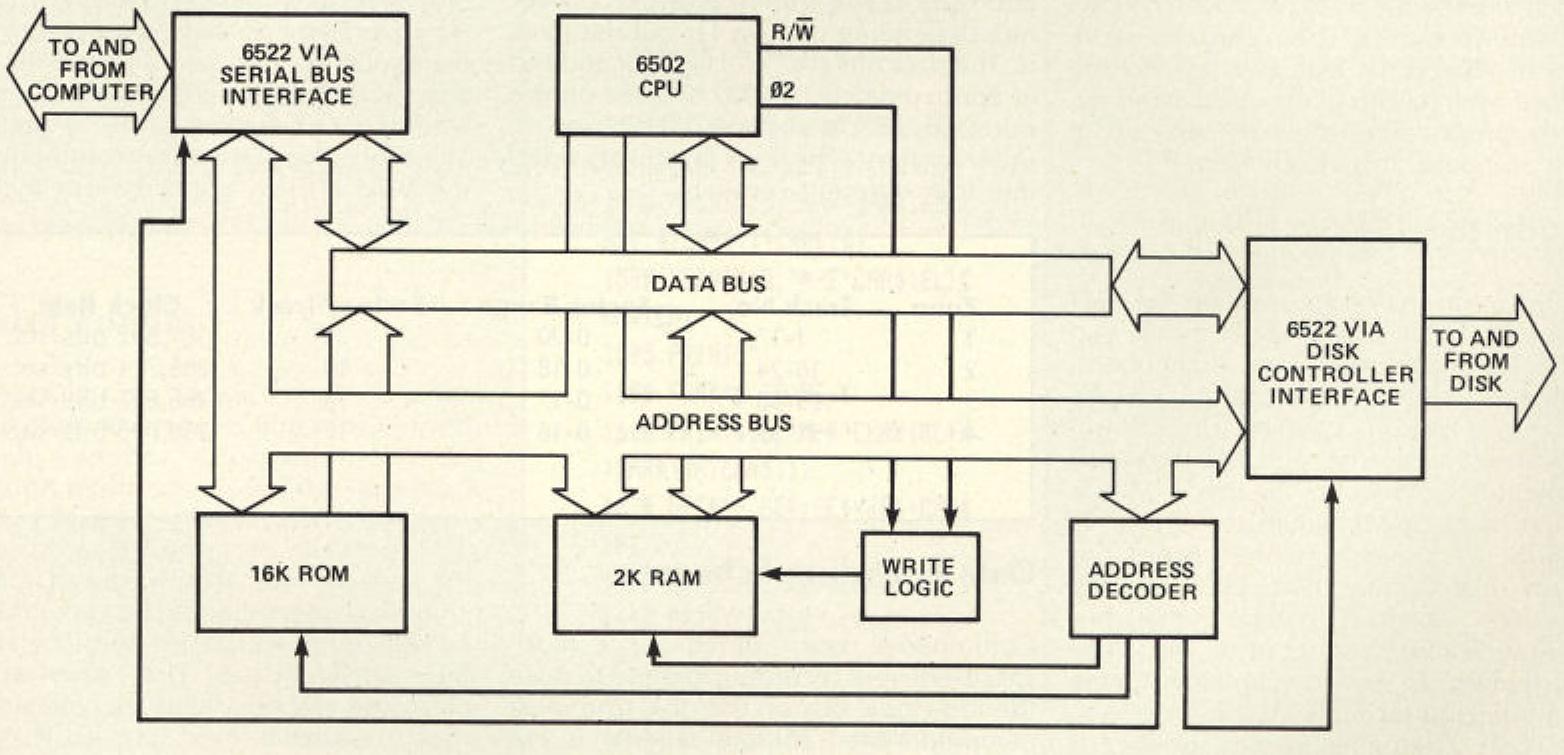


Figure 1:

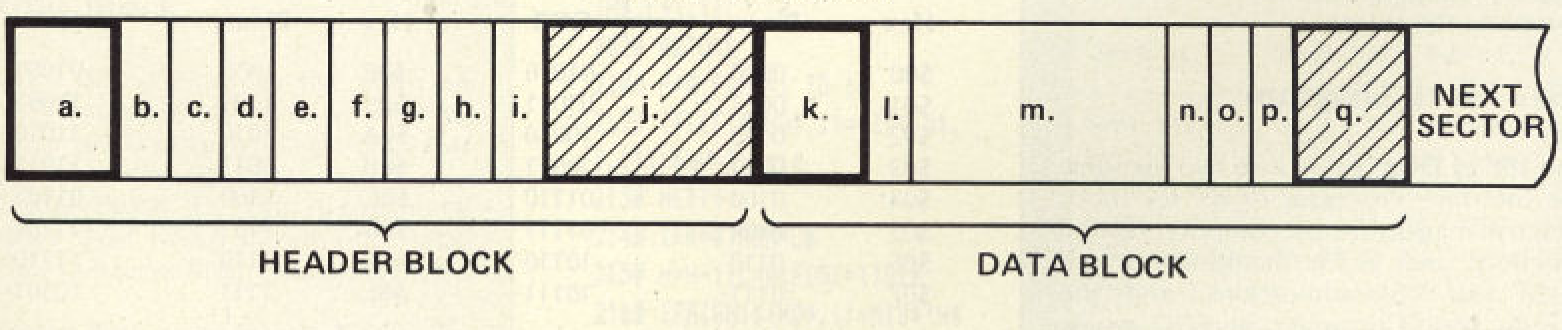


Figure 2:

versatility of supporting software, data storage is, on the whole, less efficient, and incompatability problems often arise with third party and tape software.

The 1541 Disk Drive evolved from the 1540, which was designed for the Vic-20, and, because of the limited amount of memory in the Vic, Commodore had to find a way around a memory based DOS. So was born the 1540, an intelligent serial-bus device which controlled all its own formatting, reading and writing, as well as intricate sequential and random-access file handling. Unfortunately, it was serial (slooowww), not parallel (fast), and it had quite a few bugs (it sometimes thought it was a dual drive), but, after all, anything was better than tape. When the 64 arrived on the home computer scene, it was decided a new disk drive should follow with it, but as the Vic was still selling, the new drive would have to be compatible with that too. So was born the 1541, still slow, still with bugs, but relatively cheap, quite efficient, and pretty reliable.

Because the drive is intelligent, and thus self-supporting, there is no need for either disk controller or dedicated disk software inside the 64. The control unit inside the 1541 is a 6502 based computer much like the Vic or 64, with RAM, ROM and Interface chips. This means that at the end of your serial lead is a fast storage device, which, with a little understanding, can be programmed much the same as the host computer it is attached to.

Inside the 1541

If you are prepared to open up the 1541 you will find relatively little inside that looks like a computer. The circuit board, sitting atop the mains transformer and the drive mechanism, contains the control electronics at the front end, and the actual computer circuitry at the back. This is much more Spartan than the Vic or 64 boards, as it needs neither audio nor visual related chips. Two 6522 Versatile Interface Adaptors handle serial bus communication and control the drive mechanism, as well as supplying timing and interrupt facilities for the processor. The Disk Operating System software is held in two 8K ROM chips, and 2K of RAM not only provides the necessary workspace for the 6502, but is also used as buffer storage. A handful of other support chips, including an address decoder, complete the quota.

1541 Block Diagram

The 16K of DOS is split into two sections: the Interface Processor or IFP for short, which manages the host computer related functions such as file manipulation and serial bus communications, and the Floppy Disk Controller or FDC which controls the Read-Write head and data storage. The 6502 Processor has to share time between the FDC and the IFP, which often reduces the effective operating speed of the 1541. The FDC, IFP and Interface Chips will all be elaborated on individually as the series progresses, together with a full RAM memory map.

Disk Format

In order for the DOS to find its way around the disk a format routine is used to divide the disk surface up into tracks and sectors. 35 tracks are formatted in all, track one being the first and outermost, and track 35 the innermost, with the directory on track 18. Each Track is further divided up into 17 or more 256 byte sectors, numbered zero upwards. To pack as much data into the available space, Commodore adopted a scheme where the number of sectors on a track increases the further out, and thus longer, the track is. However, this method by itself would not have worked because even if the Read-Write head is positioned on track one or 35, it still takes the same amount of time for the disk surface to rotate once. The larger the track, the higher the velocity it has when it passes under the head, so to counter this the data is actually written and read to and from the disk at a faster rate depending on how far out the track is. The data bits are "clocked" in and out at approximately 300,000 bits/sec on the outermost tracks and 250,000 bits/sec on the innermost. The track layout is divided into four different zones:

Zone	Track No.	Sector Range	Sectors/Track	Clock Rate
1	1-17	0-20	21	307,692 bits/sec
2	18-24	0-18	19	285,714 bits/sec
3	25-30	0-17	18	266,667 bits/sec
4	31-35	0-16	17	250,000 bits/sec

Data Encoding Scheme

Commodore again opted for a more space-efficient recording method to store the individual bits on the disk. The most commonly-used storage scheme is FM (Frequency Modulation) which involves writing a stream of clocking bits, with a data bit occurring in between if a "1" needs to be written. This is expensive in data storage space and so the 1541 was designed to use a self-clocking method called Group Coded Recording, GCR for short (Apple micros also use GCR). Prior to being written on to the disk, every data byte is converted into a form which can neither be confused as a sync mark nor affect reading accuracy. This is achieved by splitting the byte into two halves, or four bit nibbles, and using a look-up table in the ROM to convert each half into a five bit result.
For example, to convert the eight bit byte $A5 (10100101) into GCR, the byte is first split into two four bit hi-lo order nibbles 1010 and 0101. Using the conversion table these nibbles now become 11010 and 01111 respectively, and so our completed GCR byte is 1101001111. Using Group Codes Recording, no combination of any five bit GCR nibbles will ever produce nine consecutive binary ones (used as a sync mark), and no more than two consecutive binary zeros will appear in a 10-bit GCR byte or combination of bytes (this is for speed accuracy when clocking bits back into the 1541 during a read). However, we now have a problem when manipulating this data. The 6502 can address only one eight bit byte at a time, whereas our new byte is now 10 bits long. Therefore the conversion routine inside the DOS actually converts four bytes at the same time. This is the minimum amount of bytes that can be converted by an eight bit processor (4 * 10-bit bytes = 40 bits = 5 * 8-bit bytes). Thus, when writing data, four eight bit bytes are collated and then converted into four GCR 10-bit

GCR Table

Hex	Binary	GCR	:	Hex	Binary	GCR
$00	0000	01010	:	$08	1000	01001
$01	0001	01011	:	$09	1001	11001
$02	0010	10010	:	$0A	1010	11010
$03	0011	10011	:	$0B	1011	11011
$04	0100	01110	:	$0C	1100	01101
$05	0101	01111	:	$0D	1101	11101
$06	0110	10110	:	$0E	1110	11110
$07	0111	10111	:	$0F	1111	10101

bytes, and written as five eight bit bytes. This all sounds a bit confusing (if you'll pardon the pun) so I'll show you an example:

4 8-bit bytes : $08, $AA, $01, $01
in Binary : 00001000, 10101010, 00000001, 00000001

4-bit nibbles : 0000, 1000, 1010, 1010, 0000, 0001, 0000, 0001,
in 5-bit GCR : 01010, 01001, 11010, 11010, 01010, 01011, 01010, 01011,

in 8-bit GCR : 01010010, 011101010, 10100101, 00101101, 01001011
in hex : $52, $75, $A5, $2D, $4B,

Thus our original four eight bit bytes $08-$AA-$01-$01 are actually written on to the disk surface as $52-$75-$A5-$2D-$4B. Reading GCR bytes off the disk is merely the same process reversed.

5 GCR bytes : $52, $75, $A5, $2D, $4B,
in 8-bit GCR : 01010010, 01110101, 10100101, 00101101, 01001011

in 5-bit GCR : 01010, 01001, 11010, 11010, 01010, 01011, 01010, 01011
4-bit nibbles : 0001, 0000, 1010, 1010, 0000, 0001, 0000, 0001,

in Binary : 00010000, 10101010, 00000001, 00000001
4 8-bit bytes : $08, $AA, $01, $01

Simple, really!

Sector Format

Each sector on a track is comprised of two main parts: the Header Block, or ID Field, which supplies information on the position of the sector, and the Data Block of data. Preceding each of these fields is a unique synchronisation Field or Sync Mark, used to identify the beginning of the block and immediately following the block is a short gap which gives the FDC breathing space to allow for fluctuations in drive speed. The sector header is written only once, during formatting, but the data field, including its sync mark, is re-written every time data needs to be recorded on the disk.

Header Block

a) Sync Mark: Written as five eight bit GCR $FF bytes (40 consecutive one bits) the sync mark is a flag to tells the DOS that a block is coming up next.
b) Header Block ID: This is the Block Identifier Byte which informs the DOS that this is a header field. Its value is always $08.
c) Header Block Checksum: This is the header field checksum byte created by EORing together the track number, sector number and the two IDs.
d) Sector Number: Numbered consecutively from zero upwards.
e) Track Number: Position of the track on the disk.
f-g) ID2 and ID1: These are the formatting IDs specified in the instruction "N0:NAME,ID" where I is ID1 and D is ID2 (Note: ID2-ID1) is the correct order as written on the disk). These are the IDs that the DOS uses the Initialisation and during all read and write operations, not the "cosmetic" IDs found in sector 18,0.
h-i) $0F Bytes: These are filler bytes used as padding when the DOS is converting the header from or into GCR (Remember: the DOS uses four eight bit bytes at a time for GCR conversion). These bytes are never referenced again by the DOS after formatting.
j) Header Gap: eight eight bit GCR $55 (01010101) bytes providing the DOS with breathing space between the header and data fields.

```
1000 H$="0123456789ABCDEF"
1010 PRINT"(1) HEX TO GCR"
1020 PRINT"(2) GCR TO HEX"
1030 PRINT"(3) QUIT"
1040 GETI$:IFI$=""THEN1040
1050 I=VAL(I$):ONIGOTO1100,1
200,1070
1060 GOTO1040
1070 END
1100 NS=4:NG=NS+1:EL=208:EH=
246:WL=82:WH=00:RL=00:RH=03
1110 GOTO1500
1200 NS=5:NG=NS-1:EL=230:EH=
247:WL=00:WH=03:RL=82:RH=00
1500 OPEN15,8,15
1515 FOR K=0 TO NS-1
1520 PRINT"BYTE#";K+1;:INPUT
HD$
1530 IFLEN(HD$)<>2THENGOTO15
20
1540 GOSUB2100
1550 PRINT#15,"M-W"CHR$(WL+K
)CHR$(WH)CHR$(1)CHR$(D)
1560 NEXTK
1570 PRINT#15,"M-W"CHR$(48)C
HR$(0)CHR$(2)CHR$(0)CHR$(3)
1580 PRINT#15,"M-W"CHR$(52)C
HR$(0)CHR$(1)CHR$(0)
1590 PRINT#15,"M-E"CHR$(EL)C
HR$(EH)
1595 PRINT""
1600 FORK=0 TO NG-1
1610 PRINT#15,"M-R"CHR$(RL+K
)CHR$(RH)CHR$(1)
1620 GET#15,GE$:GE$=GE$+CHR$
(0)
1630 D=ASC(GE$):GOSUB2000
1640 PRINT"BYTE#";K+1;" $";H
D$
1650 NEXTK
1660 GETI$:IFI$=""THENGOTO16
60
1670 RUN
2000 H=INT(D/16):L=D-(H*16)
2010 HD$=MID$(H$,H+1,1)+MID$
(H$,L+1,1)
2020 RETURN
2100 H=0:FORI=1 TO 16
2120 IFLEFT$(HD$,1)=MID$(H$,
I,1)THENH=I:I=16
2130 NEXTI
2140 IFH=0THENRUN
2150 H=H-1:L=0:FORI=1TO16
2160 IFRIGHT$(HD$,1)=MID$(H$
.I.1)THENL=I:I=16
```

Data Block

k) Sync Mark: Warns the DOS a block is coming up.
l) Data Block ID: Informs the DOS that this is a Data Block. Its value is always $07.
m) Data: 256 Bytes of user data.
n) Data Block Checksum: This the data field checksum created by EORing all the 256 Bytes of data together.
o, p) $00 Bytes: Filler bytes used as padding during GCR conversion.
q) Inter-Sector Gap: This is also known as the Tail Gap, and its size varies between four and 12 eight bit GCR $55 bytes, supplying the DOS with space in between each sector to allow for fluctuations in drive speed. Its size is determined during the format sequence, which contains a routine which times how long the disk takes to rotate once. The gap size is calculated from this timing, and so explains not only why the format routine takes so long, but also why "Fast Formatters" are occassionally unreliable. Commodore used fixed gap formatting on earlier drives but found that the last sector on the track sometimes over-wrote the first slightly. Fast Formatters use fixed-gap formatting (usually eight bytes long).

Our sector is now larger and much more complicated than at first sight. Its actual size is calculated like so:

Section	8-bit Hex	8-bit GCR
Sync Mark 1		5 bytes
Header	8 bytes	10 bytes
Header Gap		8 bytes
Sync Mark 2		5 bytes
Data Block	260 bytes	325 bytes
Data Gap		4-12 bytes
	TOTAL	357-65 bytes long

Your COMMODORE

YOUR BEST INDEPENDENT COMMODORE MAGAZINE

SO YOU OWN A COMMODORE?

SO YOU'VE WRITTEN SOME PROGRAMS?

SO WHY HAVEN'T YOU SUBMITTED THEM TO US?

YOUR COMMODORE IS ALWAYS ON the lookout for new material for publication and we know that there are thousands of intelligent, literate, innovative and creative Commodore owners out there, so why don't we get together?

If you have written a program, be it a game or a utility, that you think you should share with other Commodore owners, then why not submit it to us for publication. Don't forget we pay very well for any articles that we publish.

All submissions should be well documented and where possible include a line by line breakdown of the program together with a list of any variables used. A copy of the program should be included on either tape or disk.

All submissions should be sent to the address below. If it is not chosen for publications then it will be returned to you.

You may not have written any software yourself, but you may have very firm opinions about the world of Commodore and all its attendant industries and products. If you do, then put your views or questions on paper and post them to us again at the address below – you might even get paid for airing your views!

All submissions should be sent to:

The Editor
Your Commodore
Argus Specialist Publications
No 1 Golden Square
London W1R 3AB

Office Use Only

Surname: ....

Firstnames: ....

Address: ....

....

Postcode: .... **Telephone Number:** ....

Times to contact you: ....

Program Name: ....

Computer/memory size: ....

Extras required (disk etc): ....

Have you sent the program to another magazine? Yes/No

PLEASE COMPLETE IN BLOCK CAPITALS

Date received: .... **Date ack. sent:** ....

Date checked: .... **Checked by:** ....

Rating: ....

Comments: ....

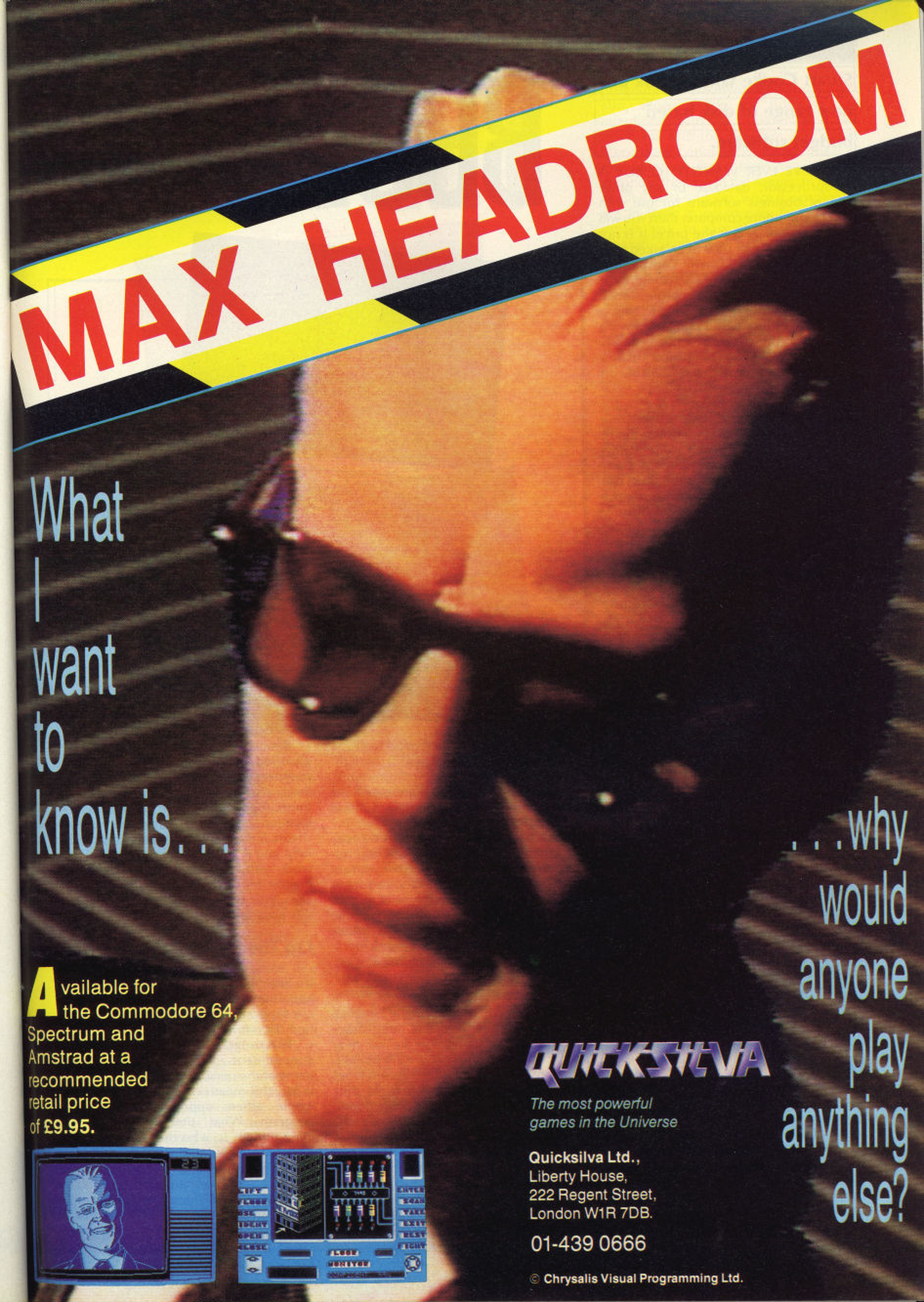
....

Reply sent: .... **Ack. received:** ....

Date printed: .... **Date returned:** ....

Fee: ....

Date Fee sent: ....



MAX HEADROOM
What I want to know is...
...why would anyone play anything else?
Available for the Commodore 64, Spectrum and Amstrad at a recommended retail price of £9.95.
QUICKSILVA
The most powerful games in the Universe
Quicksilva Ltd., Liberty House, 222 Regent Street, London W1R 7DB.
01-439 0666
© Chrysalis Visual Programming Ltd.

BUSINESS

Stuart Cooke takes a look at a new range of disk based budget software.

IF YOU'VE EVER WANTED TO BUY A wordprocessor, database or any other type of business software for use with your Commodore computer then you will have been amazed at the price. It is not unheard of for over £100 to be asked for a wordprocessor. A few companies, notably Supersoft and Mastertronic, have produced a few cheap, 'professional' programs at around the £15 mark. Now a new range of disk based budget software is set to hit the streets at a price of only £7.99. This makes it far cheaper than a great deal of software that is available on cassette.

So what's the drawback? A disk for only eight quid must have at least one fault. Well, there are a few corners cut in getting this software out so cheaply. There is no fancy packaging. A clear plastic pack is used so that the disks can hang on pegs in shops. There are no manuals with the software, the instructions on the packet simply tell you how to load the program. Finally, most of the programs are in Basic.

Don't let any of the above factors put you off the software though. The lack of instructions is usually – I say usually because one important program has no instructions – overcome by very clear help options within the program. Even though many of the programs are written in Basic they all work extremely well and do what they should. In fact many of the programs in the range work better and look better than their more expensive competitors.

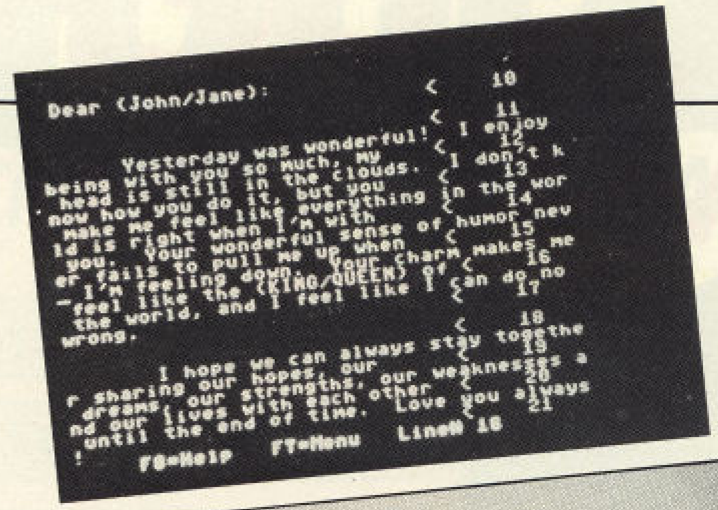
The name of this new range of software is Load'N'Go! The range is imported over from America. In the States the programs sell extremely well in supermarkets with a price tag of around $5. Who knows, we may soon be able to buy a Wordprocessor at the same time as our cornflakes.

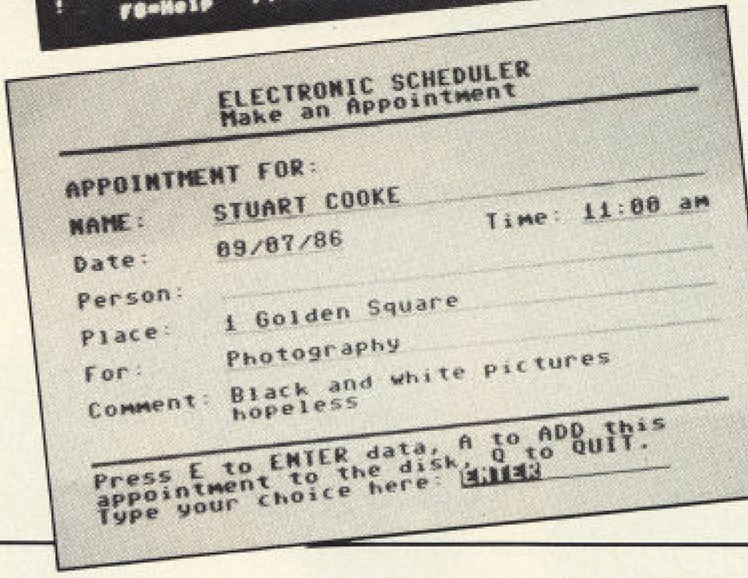
Load'N'Go! software falls into a number of different series. The series that is being marketed over here at the moment is the 'Home Management Series'. A lot of this range of software could be put to good use in either a small business or at home, its main purpose is to make life easier for you.

So what's the software really like? The best way to answer this question is to have a look at some of the packages individually.

Love Letters

At £7.99 Master Word must be the cheapest wordprocessor available. It has many of the functions of its dearer companions such as search and replace and the ability to set line spacing and margins. It does however lack the more 'up market' fancy bits such as page numbering and headers and footers. Another





Wordprocessor is due to be released at a later date that will have all of these facilities plus many more.

As wordprocessors go this one isn't particularly fancy. It will however regurgitate your latest masterpieces without complaining at all. In fact if you're not too good with the old pen a selection of Business letters (You're fired etc), Home letters (birthday etc) and love letters (Yuck!!!) are provided on the disk.

Adding it up

Another program in the series is 'Personal Spreadsheet'. This one is a little disappointing. For a start there are no clear instructions anywhere. There are no help functions, except with the different types of calculation. This means that the only way to figure out what the package does is to actually use it. If you have never come across a spread sheet before then I suggest that you find out what one is and how it works before you look at this program. If you have already used a spreadsheet then you will more than likely be disappointed with this package.

Basically 'Personal Spreadsheet' is a glorified calculator. You can store numbers in the columns and rows and then perform simple calculations on them. You cannot set up formulae in locations as you can on other spreadsheets. Since a spreadsheet is supposed to help you if you have a lot of formulae to work with, I think this program is the biggest let down of the pack. A normal calculator will perform the same job as this program, and a lot quicker.

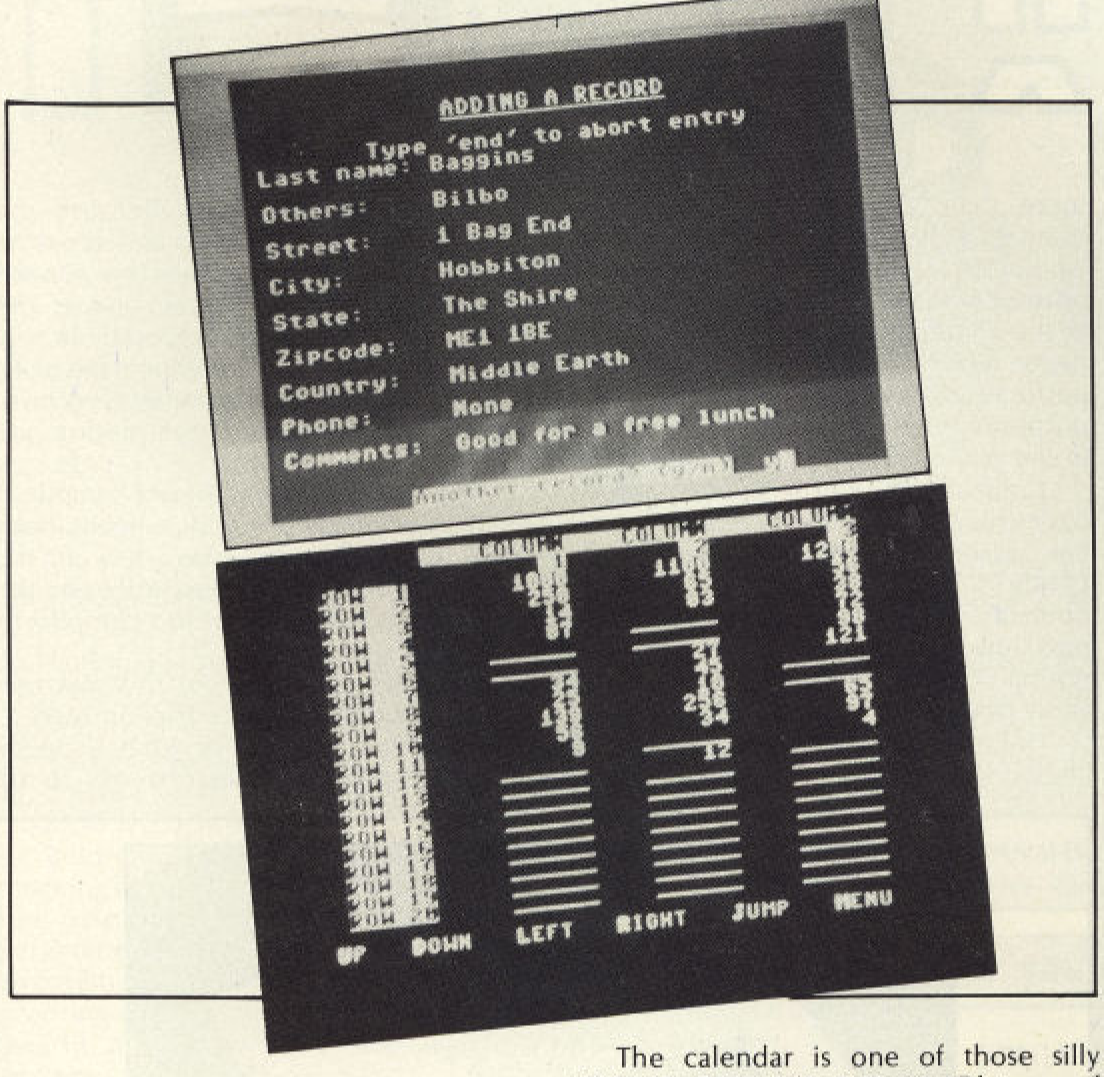
On Time

If, like me, you are always forgetting appointments or forgetting what time you are supposed to be somewhere, you will find 'Electronic Schedular' a must.

This program is used for recording all appointments. You are asked for the name of the person who the appointment is for, this means that you could put the whole offices appointments into this program. What time and date the appointment is, who the appointment is with, where it is and any comments that you need.

Once you have entered in the data you can search for all meetings for a certain person after certain dates. You can list all

On a BUDGET



meetings between dates. It is even possible to print details of all meetings at a certain place or with certain people.

How on earth did I ever manage without this one?

Money Matters

A couple of financial organisers are included in the series. These are 'Pro Financial Organiser' and 'Home Finance Organiser 1'. Both disks offer checkbook and address list programs while the Financial organiser also has a calendar and the Home finance package has a budget program.

One very nice feature in the Mail List manager on the Pro disk is the ability to code names and addresses, with say an 'F' for friend or 'B' for business. When you want to print out your labels you can then print them out for just one group.

A very nice 'extra' is the Christmas card list. For each record you must specify whether you want them on your christmas card list or not. You can then get a count of how many Christmas cards you will need and then get the computer to print out all the labels.

The calendar is one of those silly 'PLEASE GIVE ME THE YEAR' type of programs. You know, you give it the year and the month and it prints out a calendar for you. One extra feature is the ability to highlight a specific date. This would be great if you could highlight all dates with say birthdays of friends but since you can only highlight one day in every month this is probably out of the question. What a silly fault!

The titles explain the functions of the other programs on these disks and I don't think that I need say anything more about them apart from that they work well. In fact if you used either of the chequebook programs you would know exactly how your finances were at any time.

Both disks are good in certain areas. If you want addresses then go for the Pro Finance pack, if you want budgeting and cheques then I think that the Home finance pack is a little better.

The packs are so cheap that it may well be worth buying both.

More Info

If you require more information with your address file then you could have a look at the Home and Business Card File disk. This is very similar to the address programs that are mentioned above but it also allows fields for business and telephone numbers. I did have a few problems with this program. For a start there wasn't enough room for many of the addresses that I tried to enter and secondly they are in the American format of City, State and Zip. Since the programs are in Basic it would only have been a simple task to turn these formats into 'English'. It may even be possible to make the changes yourself.

Even better than this is 'g' base. This is a proper database program. By proper, I mean that you can define your own fields for data entry. This means that if you wanted to make an address list you aren't stuck with ZIP etc. If you wanted you could even use the program to keep a catalogue of your records or stamps. In fact 'g' base can be used to store information about anything that you could put on a card. Having the information on computer means that it is a simple task to find specific details or print out a list etc.

Keeping Tabs

Obviously with disk software becoming so cheap the number of disks in your collection is bound to grow. How on earth are you going to keep track of them all?

Well, 'Disk Utilities' will solve this problem for you. This program will store information on up to 300 disks. You can search for a specific program and you will be told what disk it is on. You can get a print out of the directories of all your disks. It is even possible to print out a list of all the IDs that you have used. As you are probably aware the Commodore disk drive uses a two digit ID to identify each disk. If two disks have the same ID, the disk drive may not know if you swap the disk, thus messing up the contents.

The list of used IDs is printed in the form of a grid with letters and numbers being across the top and edge of the grid. It is now easy to mark off new disks as you format them.

As well as the catalogue program there is also a utility program. This will allow you to backup disks (not protected ones), format disks etc.

At a price of only £7.99 this disk is a must for any disk drive owners.

Verdict

Most of the programs available are around the same quality as a good magazine listing. They are all functional and do what they set out to do without any frills.

At £7.99 this software should make a very big impact on the home/small business market, perhaps with the same sort of impact that Mastertronic made when it launched its £1.99 tapes.

Who knows, good business software at this price may make people start to use their computers for something other than games, this wouldn't be such a bad thing.

IN PRINT

Eric Doyle has been trying out a real hardware bargain.

WHEN I READ THAT THE IBICO LTR-1 printer claimed letter quality printing and that the price was less than £100, my immediate reaction was, to say the least, one of disbelief. After unpacking the beast I remained unimpressed, the printer head looked very Mickey Mouse-ish but when I tried printout my attitude changed.

Surprisingly, the machine does produce letter quality and it does so in a very novel way. The printer head resembles an office date stamp. You know the kind, rotate the wheels to get the correct date, then ink the rubber-faced letters on a pad and stamp away. In this novel little printer the characters are carried on a cylinder which carries four bands of characters. At the back of the wheel is an inked cylinder which brushes against the letters as they rotate and each character is pressed against the paper when required to produce very high quality letters. High-tech still has a place for the good old principles pioneered by Guttenberg!

The limitation of the printer lies in the fixed roller idea. The characters you get are the ones you're stuck with. No graphics screen dumps or characteristic Commodore symbols, just plain and simple alphanumerics and punctuation.

The paper is friction fed through the printer with no facilities for tractor feeding. This means that as a cheap printer for wordprocessing applications it can use high quality, headed paper. Indeed, a good wordprocessor would be a boon to anyone owning this machine because it doesn't have a 'paper out' indicator. When the printer reaches the bottom of a page, the platten loses its grip on the paper and the head carries on printing to and fro across the same line. A wordpro with the facility to stipulate a page length would overcome this fault.

For most people the main application they would look for is the ability to print out listings of their latest project for a leisurely debugging session. Obviously the problem here is that most listings fill more than a sheet of A4 paper and estimating how many lines to a page would be a nightmare. I found that normal A4 tractor feed paper would not fit the platten mounting but you can buy paper which is A4 width including the perforations and this is fine for most purposes as long as the paper doesn't slip in the platten rollers.

Control of the printer is limited to the basic character codes of the Commodore. For example, CHR$(10) followed by CHR$(13) will initiate a line feed and home the printer to the beginning of the next line. CHR$(28) sets the printer to normal line spacing and CHR$(29) will allow double spacing for extra clarity.

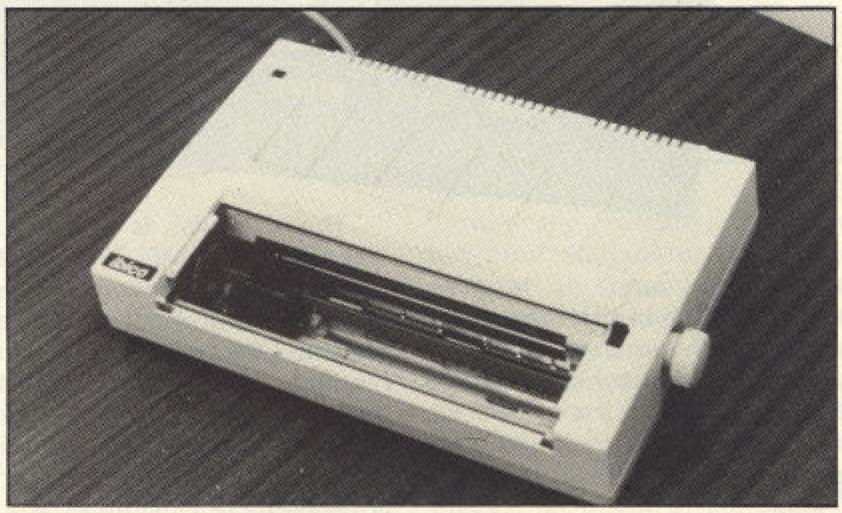
The number of characters available is limited to 126. This includes all the alphanumeric characters and punctuation marks. In the main these correspond to Commodore's version of ASCII but some of the characters around CHR$(90) vary. The most important difference is that the pound sign has an ASCII value of 123 instead of 92 but a little bit of thought should overcome this problem.

Despite the Heath Robinson appearance of the printer, the quality of the printout is excellent. After a while the letters may get a little faint or the entirity of their outlines become broken but this can be remedied by the simple application of more ink on the roller giving a result indistinguishable from the quality of a typewriter.

The speed of the printer is comparable to that of a daisywheel. Because the printer face has to move up and down the paper frequently, this means that a speed of 10-12 characters per second is the maximum that can be achieved. In real terms this means that a 60-line page of A4 text will take about five minutes to print which is not too bad considering the quality.

The total size of the printer is about 12 inches by nine by two, which means that it takes up very little room on a desk and the only control is the paper advance and the on/off switch so it is not too complex to use.

My only qualms about this machine are the tendency of the friction feed to slip and the problem of what happens when the print head wears down. To be fair, mine is a review machine which may have seen hard use at the hands of others and the resilience of the printer head seems to promise a fairly long life.



The cost of the printer should be seen in the light of the need for a centronics interface. I believe Boots sell one for around £20, so if you're in the market for a cheap printer with high quality result and you're willing to accept the 20th Century adage that budgeters can't be choosers then this is definitely a machine to consider. The only other printers at this price are thermal printers and with the cost of thermal paper these days it could be worth while considering a machine with low post-sales overheads, in which case this could be the one for you.

BUDGET 64

David Gartrell helps you get your finances in order with the help of your C64.

IF YOU'RE TIRED OF wondering just how much money you've got in the bank and whether you're going to last until your next pay packet comes through, then look no further. This program can be used to monitor all of your incomings and out-goings so that you, and not your bank manager, are the first to know where you stand, financially speaking.

Introduction

Using Budget 64 you can divide your bank account into a maximum of 20 separate categories, each covering a different section of your finances e.g. gas, electricity.

Before we delve into the workings of the program, there are one or two things to remember. The structure of the program is such that, when it is working at full capacity, it uses nearly all of the available memory. Only 2-3K remains. Also, from time to time, pauses will occur. This does not mean that the computer has crashed. Just wait for a few seconds and it will carry on. The delays are caused by the complexity of the program, the many variables used and the amount of memory involved.

When the program is first run, you are presented with a title screen and a question: 'LOAD OLD FILE Y/N'. If you are starting afresh, then type 'N'. Details of loading your file will be explained later.

Next the computer will ask you how many categories you wnat. These are the different sections of your bank account. Enter the number you want plus one, then enter their names.

The first category is pre-set as 'MASTER SHEET' and combines all the others to give a grand balance.

Once all the names have been entered, the main menu is displayed. Pressing a key will display up to three pages of transactions for each category. You have the choice of display on the screen, the 1520 Printer Plotter or a standard printer. The printer used when this program was written was a Star SG-10C dot matrix printer. However, using other printers should cause no problems.

Included in the display are reverse characters to identify the origin of each transaction. Press the key followed by the character of the transaction which you wish to identify.

At the bottom of the main menu is 'OTHER OPTIONS'. These are:

ACTIVATE ACCOUNT: Enables you to add or subtract from any of your categories. The process is straightforward and simple.

SAVE A FILE: Will save your file on tape. The option of an access code is included to prevent other people from loading your file.

ENTER WAGE: This option can be used if, for instance you wanted to divide £100 between different categories. Enter the total amount deposited and press return. The amount will be displayed on the other options menu and any other areas where is could be useful. As you spread your money between the categories, the amount you entered will decrease. When it reaches zero it will disappear.

Options E and F work together with option D. If you remove the messager from the screen, the amount will remain the same until you bring it back.

RE-NAME CATEGORY: Choose the category which you wish to re-name. Make sure that it is empty of money as its memory will be cleared when you re-name it.

MONEY TRANSFER: Enter the two categories involved, then proceed as for 'ACTIVATE ACCOUNT'.

ADD A CATEGORY: Type in the name of the new category.

PERFORM CALCULATIONS: If you want to do any arithmetic you can use this option.

Finally to load your file, press 'Y' at the start. Then enter your access code if you have one.

PROGRAM: BUDGET64

```
1 PRINT CHR$(14):POKE 53280,9:POKE 53281,11
2 PRINT"[CLEAR,C1,RVSON,SPC16,RVSOFF,YELLOW,SB,SU,SD,SG,
   SE,ST]-64[RVSON,C1,SPC15]"
3 GOSUB 7000
4 PRINT"[DOWN4,YELLOW,SPC7,SW]RITTEN BY[SPC,SD]AVID[SPC,
   SG]ARTRELL"
5 PRINT"[DOWN,SPC11,SC]OPYRIGHT (C) 1985"
6 PRINT"[DOWN4,BLACK]LOAD OLD FILE? :-[SPC5,SY]/[SN]"
7 A$="":GET A$:IF A$=""THEN 7
8 IF A$="N"THEN OPEN 1,3:GOTO 24
9 IF A$<>"Y"THEN 7
10 PRINT"[CLEAR,C1,RVSON,SPC15,RVSOFF,YELLOW,SL]
   OAD A FILE[RVSON,C1,SPC14]"
11 PRINT"[UP,RED,CT40,C6]"
12 PRINT"[DOWN2,SE]NTER ACCESS CODE :-[C3,RIGHT]";:AC$=""
13 GET C$:IF C$=""THEN 13
14 IF C$=CHR$(13)THEN 16
15 AC$=AC$+C$:PRINT"*";:GOTO 13
16 AC$=AC$+"[CLEAR]"
17 OPEN 1,1,0,AC$:DIM A$(21),DA$(21,50),DE(21,50)
18 DIM WI(21,50),BA(21,50),RO%(21),XX$(21,50)
19 PRINT"[HOME,DOWN3]":FOR I=1 TO 20:PRINT"[SPC39]":NEXT
20 INPUT#1,A
21 FOR B=1 TO A:INPUT#1,A$(B):INPUT#1,RO%(B)
   :FOR BB=1 TO RO%(B)-1:INPUT#1,DA$(B,BB)
```

```
22 INPUT#1,DE(B,BB):INPUT#1,WI(B,BB):INPUT#1,BA(B,BB)
   :INPUT#1,XX$(B,BB):NEXT BB,B
23 CLOSE 1:OPEN 1,3:GOTO 35
24 PRINT"[CLEAR,C1,RVSON,SPC16,YELLOW,RVSOFF,SB,SU,SD,SG,
   SE,ST]-64[C1,RVSON,SPC15]"
25 GOSUB 7000
26 PRINT"[DOWN3,BLACK,ST]HE MAXIMUM NUMBER OF CATEGORIES
    YOU CAN"
27 PRINT"[UP]HAVE IS TWENTY."
28 PRINT"[DOWN2,CYAN,SN]O. OF CATEGORIES":INPUT A
   :IF A<1 OR A>20 THEN PRINT"[UP,SPC12,UP4]":GOTO 28
29 DIM A$(21),DA$(21,50),DE(21,50),WI(21,50),BA(21,50),
   RO%(21),XX$(21,50)
30 A$(1)="[SM]ASTER SHEET":FOR B=1 TO A:RO%(B)=1:NEXT
31 PRINT"[DOWN2,SC]ATEGORY 1 HAS BEEN AUTOMATICALLY SET
    AS:-[SPC,RVSON,SM]ASTER SHEET"
32 IF A=1 THEN 35
33 PRINT"[DOWN2,C6]":FOR B=2 TO A:PRINT"[SC]ATEGORY "B"
   :-[DOWN,C7]":INPUT A$(B)
34 PRINT"[UP,SPC36,UP3,C6]":NEXT
35 PRINT"[CLEAR,C1,RVSON,SPC16,RVSOFF,YELLOW,SM,SA,SI,SN,
   SPC,SM,SE,SN,SU,RVSON,C1,SPC15]"
36 PRINT"[RED,UP,CT40,WHITE]"
37 DEF FNR(KL)=INT(KL*100+.5)/100
38 S=1:GOSUB 112
39 PRINT TAB(11)"[C8]"CHR$(A+65)" = OTHER OPTIONS."
40 PRINT"[WHITE,RVSON,SC,SH,SO2,SS,SE,RVSOFF]:-"
41 B$="":GET B$:IF B$=""THEN 41
42 X=ASC(B$)-64:IF X<1 OR X>A+1 THEN 41
43 IF X=A+1 THEN 114
44 PRINT"[CLEAR,DOWN2,SW]HICH PAGE :- 1, 2 OR 3"
45 GOSUB 355:PRINT"[DOWN,SC]URRENT PAGE :-"A1
46 GET A1$:IF VAL(A1$)<1 OR VAL(A1$)>A1 THEN 46
47 GOSUB 350
48 A3=A2+15:IF RO%(X)<=A3 THEN A3=RO%(X)-1
49 PRINT"[CLEAR,DOWN3]DISPLAY ON :-"
50 PRINT"[DOWN]3 = SCREEN":PRINT"4 =[SPC,SP]RINTER"
   :PRINT"6 = 1520[SPC,SP]RINTER/[SP]LOTTER"
51 GET Y$:IF Y$=""THEN 51
52 IF Y$="4"THEN CLOSE 1:OPEN 1,4,7:PRINT#1,CHR$(14)
   :GOTO 57
53 IF Y$="3"THEN GOTO 57
54 IF Y$<>"6"THEN 51
55 CLOSE 1:OPEN 1,6
56 OPEN 6,6,6:PRINT#6,1:OPEN 2,6,1
57 GOSUB 207
58 IF Y$="3"THEN PRINT#1
59 PRINT#1:PRINT#1,"[BLACK,UP3,SPC2]DETAILS[SPC4]DEPOSIT
   [SPC2]WITHDRAWL BALANCE"
60 PRINT#1," ---------------------------------------"
61 IF Y$="4"THEN PRINT#1,CHR$(15):CLOSE 1
62 GOTO 68
63 FOR P=1 TO RO%(X):DA$(X,P-1)=DA$(X,P)
64 DE(X,P-1)=DE(X,P):WI(X,P-1)=WI(X,P)
65 BA(X,P-1)=BA(X,P)
66 XX$(X,P-1)=XX$(X,P):NEXT
67 RETURN
68 FOR B=A2 TO A3
69 IF Y$<>"4"THEN 74
70 GOSUB 321:PRINT#1," "DA$(X,B):PRINT#2,PB$
   :PRINT#3,DE(X,B)
71 PRINT#2,PC$:PRINT#3,WI(X,B):PRINT#2,PE$
   :PRINT#3,"[RVSON]"XX$(X,B)"[RVSOFF]"
72 PRINT#2,PD$:PRINT#3,BA(X,B)
73 CLOSE 1:CLOSE 2:CLOSE 3:CLOSE 6:PRINT#10:CLOSE 10
   :GOTO 89
74 P1=11:P2=19:P3=29
75 LK=BA(X,B):GOSUB 264
76 IF Y$="6"THEN GOTO 302
77 GOSUB 281
78 IF Y$="6"THEN PRINT#2,"M",12,0:PRINT#1,DA$(X,B);
   :GOTO 81
79 PRINT#1," "DA$(X,B);
80 IF Y$="4"THEN OPEN 1,3
81 IF Y$="6"THEN PRINT#2,"M",132,0:PRINT#1,P$DE$;:GOTO 83
82 PRINT#1,TAB(P1)DE$;
83 IF Y$="6"THEN PRINT#2,"M",228,0:PRINT#1,Q$WI$;:GOTO 85
84 PRINT#1,TAB(P2)WI$;
85 IF Y$="6"THEN PRINT#2,"M",351,0:PRINT#1,XX$(X,B);
   :GOTO 87
86 PRINT#1,TAB(29)"[RVSON]"XX$(X,B)"[RVSOFF]";
87 IF Y$="6"THEN PRINT#2,"M",349,0:PRINT#1,R$BA$:GOTO 89
88 PRINT#1,TAB(P3)BA$
89 NEXT
90 IF Y$="4"THEN OPEN 1,3
91 IF Y$="6"THEN FOR J=1 TO 7:PRINT#1:NEXT:CLOSE 6:CLOSE 2
   :CLOSE 1:OPEN 1,3
92 PRINT TAB(14)"[YELLOW,RVSON,SP,SR,SE,SS2] '*' [SK,SE,
   SY]"
93 PRINT"[RVSON,RIGHT6,SP,SR,SE,SS2,SPC]'@'[SPC,SF,SO,SR,
   SPC,SI,SD,SE,SN,ST,SI,SF,SI,SC,SA,ST,SI,SO,SN]"
94 GET A$:IF A$=""THEN 94
95 IF A$="*"THEN 105
96 IF A$<>"@"THEN 94
97 PRINT"[HOME,DOWN23,SPC39]"
98 INPUT"[HOME,DOWN22,SPC5,LEFT5]";K$
99 IF K$="*"THEN 35
100 IF K$=""THEN 97
101 IF LEN(K$)=1 AND ASC(K$)<65 OR ASC(K$)>A+64 THEN 97
102 IF LEFT$(K$,1)="*"THEN PRINT"[C6,ST]RANSFERED TO
    [RVSON,RIGHT]";:K$=RIGHT$(K$,1):GOTO 104
103 IF RIGHT$(K$,1)="*"THEN PRINT"[C6,ST]RANSFERED FROM
    [RVSON,RIGHT]";:K$=LEFT$(K$,1)
104 K=ASC(K$)-64:PRINT"[RVSON,C6]"A$(K)"[GREEN,YELLOW]"
    :GOTO 94
105 GOTO 35
106 PRINT"[CLEAR,C1,RVSON,SPC15,RVSOFF,YELLOW,SE]
    NTER WAGE[RVSON,C1,SPC15]"
107 GOSUB 7000
108 WW=1:YY=0:INPUT"[CYAN,DOWN3,SM]ONEY FOR DISTRIBUTION";
    YY:GOTO 114
109 WW=0:GOTO 114
110 WW=1:GOTO 114
111 END
112 PRINT"[UP2,C5]":FOR B=S TO A:B$=CHR$(B+64)
    :PRINT TAB(11)CHR$(B+64)" = "A$(B)
```

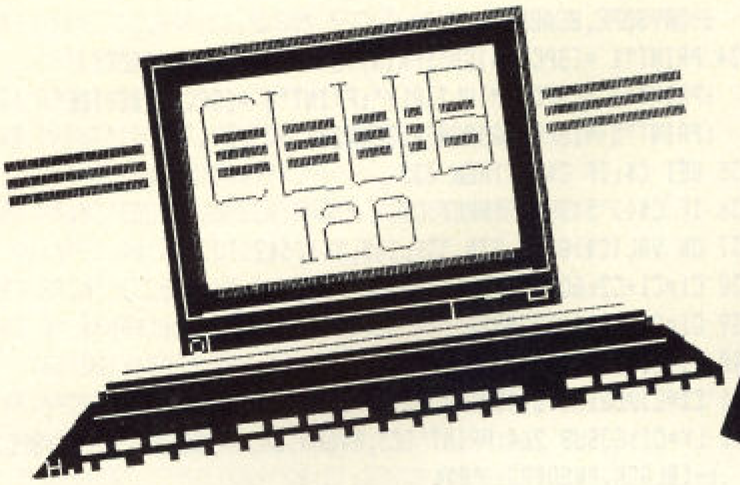
```
113 NEXT:RETURN
114 PRINT"[CLEAR,RVSON,C1,SPC14,RVSOFF,YELLOW,SO,ST,SH,SE,
  SR] [SO,SF,ST,SI,SO,SN,SS,RVSON,C1,SPC13]"
115 GOSUB 7ØØØ
116 IF WW=1 AND YY=Ø THEN WW=Ø
117 IF WW=1 THEN LK=YY:GOSUB 264:PRINT"[UP,YELLOW,RVSON,
  SW]AGE LEFT :- £"BA$
118 PRINT TAB(8)"[DOWN,C8,SPC]A =[SPC,SA]CTIVATE ACCOUNT."
  :PRINT TAB(8)" B =[SPC,SS]AVE A FILE."
119 PRINT TAB(8)" C =[SPC,SR]ETURN TO MAIN MENU."
  :PRINT TAB(8)" D =[SPC,SE]NTER WAGE."
12Ø PRINT TAB(8)" E =[SPC,SC]ANCEL WAGE MESSEAGE."
121 PRINT TAB(8)" F =[SPC,SE]NABLE WAGE MESSAGE."
122 PRINT TAB(8)" G =[SPC,SR]E-NAME CATEGORY."
123 PRINT TAB(8)" H =[SPC,SM]ONEY TRANSFER."
124 PRINT TAB(8)" I =[SPC,SA]DD A CATEGORY."
125 PRINT TAB(8)" J = RE-RUN PROGRAM."
126 PRINT TAB(8)" K =[SPC,SP]ERFORM CALCULATIONS."
127 PRINT"[WHITE,DOWN,RVSON,SC,SH,SO2,SS,SE,RVSOFF,SPC]:-"
128 B$="":GET B$:IF B$=""THEN 128
129 IF ASC(B$)-64<1 OR ASC(B$)-64>11 THEN 128
13Ø ON ASC(B$)-64 GOTO 131,151,35,1Ø6,1Ø9,11Ø,196,211,254,
  259,33Ø
131 PRINT"[CLEAR,C1,RVSON,SPC12,RVSOFF,YELLOW,SA]CTIVATE
  ACCOUNT[RVSON,C1,SPC12]"
132 GOSUB 7ØØØ
133 IF WW=1 AND YY=Ø THEN WW=Ø
134 IF WW=1 THEN LK=YY:GOSUB 264:PRINT"[UP,YELLOW,RVSON,
  SW]AGE LEFT :- £"BA$
135 :S=2:PRINT"[DOWN]":GOSUB 112:PRINT"[WHITE,RVSON,SC,SH,
  SO2,SS,SE,RVSOFF,SPC]:-"
136 GET B$:IF B$=""THEN 136
137 X=ASC(B$)-64:IF X<2 OR X>A THEN 136
138 GOSUB 2Ø7
139 IF WW=1 AND YY=Ø THEN WW=Ø
14Ø IF WW=1 THEN LK=YY:GOSUB 264:PRINT"[UP,YELLOW,RVSON,
  SW]AGE LEFT :- £"BA$
141 LK=BA(X,ROZ(X)-1):GOSUB 264
142 PRINT"[DOWN2,C7,SM]ONEY IN ACCOUNT :-£"BA$
  :IF QU=1 THEN RETURN
143 PRINT"[DOWN3,YELLOW]1 =[SPC,SA]DD TO ACCOUNT."
  :PRINT"[DOWN,C3]2 =[SPC,SS]UBTRACT FROM ACCOUNT."
144 PRINT"[DOWN,WHITE]3 =[SPC,SR]ETURN TO MENU."
145 B$="":GET B$:IF B$=""THEN 145
146 IF VAL(B$)<1 OR VAL(B$)>3 THEN 145
147 IF B$="1"THEN T$="[YELLOW,RVSON,SA,SD2]"
148 IF B$="2"THEN T$="[C3,RVSON,SS,SU,SB,ST,SR,SA,SC,ST]"
149 IF B$="3"THEN T$="[WHITE,RVSON,SR,SE,ST,SU,SR,SN]"
15Ø PRINT"[DOWN]"T$:T$="":GOTO 168
151 PRINT"[CLEAR,C1,RVSON,SPC15,RVSOFF,YELLOW,SS]AVE A
  FILE[RVSON,C1,SPC14]"
152 PRINT"[UP,RED,CT4Ø,C6,DOWN4]"
153 CLOSE 1
154 PRINT"[DOWN2,SD]O YOU WANT TO HAVE AN ACCESS CODE[SPC,
  SY]/[SN]"
155 GET W$:IF W$=""THEN 155
156 IF W$="N"THEN 158
157 PRINT"[RVSON,C6,SE]NTER ACCESS CODE :-[SPC,C3]";
  :AC$="":GOSUB 278:GOSUB 166
158 PRINT:OPEN 1,1,1,AC$
159 PRINT"[HOME,DOWN3]":FOR I=1 TO 2Ø:PRINT"[SPC39]":NEXT
16Ø PRINT#1,A:FOR B=1 TO A:IF A$(B)=""THEN A$(B)="[C4]*"
161 PRINT#1,A$(B):PRINT#1,ROZ(B):FOR BB=1 TO ROZ(B)-1
162 IF DA$(B,BB)=""THEN DA$(B,BB)="[C4]*[BLACK]"
163 IF XX$(B,BB)=""THEN XX$(B,BB)="[C4]*[BLACK]"
164 PRINT#1,DA$(B,BB):PRINT#1,DE(B,BB):PRINT#1,WI(B,BB)
  :PRINT#1,BA(B,BB)
165 PRINT#1,XX$(B,BB):NEXT BB,B:CLOSE 1:OPEN 1,3:GOTO 114
166 IF LEFT$(AC$,1)="[CLEAR]"THEN AC$=MID$(AC$,1):RETURN
167 AC$=AC$+"[CLEAR]":RETURN
168 ON VAL(B$)GOTO 169,183,114
169 PRINT"[DOWN2,CYAN,SA]MOUNT":INPUT DE(X,ROZ(X))
17Ø IF DE(X,ROZ(X))=<YY THEN 172
171 IF WW=1 THEN WW=Ø
172 INPUT"[DOWN2,GREEN,SD]ETAILS :-";DA$(X,ROZ(X))
173 WI(X,ROZ(X))=Ø:BA(X,ROZ(X))=BA(X,ROZ(X)-1)+DE(X,
  ROZ(X))
174 DA$(1,ROZ(1))=DA$(X,ROZ(X)):DE(1,ROZ(1))=DE(X,ROZ(X))
175 WI(1,ROZ(1))=WI(X,ROZ(X)):BA(1,ROZ(1))=BA(1,
  ROZ(1)-1)+DE(X,ROZ(X))
176 XX$(X,ROZ(X))=CHR$(X+64):XX$(1,ROZ(1))=CHR$(X+64)
177 IF WW=1 THEN YY=YY-DE(X,ROZ(X))
178 IF ROZ(X)=49 THEN GOSUB 63
179 IF ROZ(1)=49 THEN XX=X:X=1:GOSUB 63:X=XX
18Ø IF ROZ(X)<49 THEN ROZ(X)=ROZ(X)+1
181 IF ROZ(1)<49 THEN ROZ(1)=ROZ(1)+1
182 GOTO 114
183 PRINT"[DOWN2,CYAN,SA]MOUNT :-":INPUT WI(X,ROZ(X
  :INPUT"[DOWN2,GREEN,SD]ETAILS :-";DA$(X,ROZ(X))
184 LK=BA(X,ROZ(X)-1):GOSUB 264:IF WI(X,
  ROZ(X))=<VAL(BA$)THEN 187
185 PRINT"[CLEAR,DOWN6,RIGHT12,RVSON,YELLOW,SI,SN,SS,SU,
  SF2,SI,SC,SI,SE,SN,ST] [SF,SU,SN,SD,SS]"
  :FOR T=1 TO 15ØØ:NEXT
186 WI(X,ROZ(X))=Ø:DA$(X,ROZ(X))="[C4]*":GOTO 114
187 DE(X,ROZ(X))=Ø:BA(X,ROZ(X))=BA(X,ROZ(X)-1)-WI(X,
  ROZ(X);
188 DA$(1,ROZ(1))=DA$(X,ROZ(X)):DE(1,ROZ(1))=DE(X,ROZ(X))
189 WI(1,ROZ(1))=WI(X,ROZ(X)):BA(1,ROZ(1))=BA(1,
  ROZ(1)-1)-WI(X,ROZ(X))
19Ø XX$(X,ROZ(X))=CHR$(X+64):XX$(1,ROZ(1))=CHR$(X+64)
191 IF ROZ(X)=49 THEN GOSUB 63
192 IF ROZ(1)=49 THEN XX=X:X=1:GOSUB 63:X=XX
193 IF ROZ(X)<49 THEN ROZ(X)=ROZ(X)+1
194 IF ROZ(1)<49 THEN ROZ(1)=ROZ(1)+1
195 GOTO 114
196 PRINT"[CLEAR,C1,RVSON,SPC12,YELLOW,RVSOFF,SR]E-NAME
  CATEGORY[RVSON,C1,SPC12]"
197 GOSUB 7ØØØ
198 S=2:GOSUB 112:PRINT"[WHITE,DOWN,RVSON,SC,SH,SO2,SS,SE,
  RVSOFF,SPC]:-"
199 B$="":GET B$:IF B$=""THEN 199
2ØØ X=ASC(B$)-64:IF X<1 OR X>A THEN 199
2Ø1 ROZ(X)=1:DA$(X,1)="":DE(X,1)=Ø:WI(X,1)=Ø:BA(X,1)=Ø
  :XX$(X,1)=""
2Ø2 BA(X,Ø)=Ø
```

```
203 PRINT"[CLEAR,C1,RVSON,SPC12,YELLOW,RVSOFF,SR]E-NAME
   CATEGORY[RVSON,C1,SPC12]"
204 GOSUB 7000
205 PRINT"[DOWN2,YELLOW,SP]RESENT CATEGORY '"B$"' IS "
   :PRINT"[RVSON]"A$(X)
206 PRINT"[DOWN3,GREEN]ENTER NEW CATEGORY "B$" :-"
   :INPUT A$(X):GOTO 114
207 PRINT"[CLEAR,C1,RVSON,SPC40]"
208 Q=LEN(A$(X)):QQ=(40-Q)/2:IF Y$="4"THEN PRINT#1,
   TAB(QQ);:GOTO 210
209 PRINT"[HOME]";:FOR Q1=1 TO QQ:PRINT#1,"[RVSON,SPC]";
   :NEXT
210 PRINT#1,"[YELLOW]"A$(X):GOSUB 7000:RETURN
211 PRINT"[CLEAR,C1,RVSON,SPC13,RVSOFF,YELLOW,SM]
   ONEY TRANSFER[RVSON,C1,SPC13]"
212 GOSUB 7000
213 S=2:GOSUB 112:PRINT"[ST]RANSFER FROM CATEGORY :-";
   :INPUT C$
214 IF C$=""THEN 114
215 PO=ASC(C$)-64:GOSUB 260
216 C=ASC(C$)-64:IF BA(C,RO%(C)-1)>0 THEN 219
217 PRINT"[CLEAR,DOWN9,YELLOW,RVSON,RIGHT8,SN,SO,SPC,SM,
   SO,SN,SE,SY,SPC,SF,SO,SR,SPC,ST,SR,SA,SN,SS,SF,SE,SR]"
218 :FOR T=1 TO 1500:NEXT:GOTO 114
219 PRINT"[CLEAR,C1,RVSON,SPC13,RVSOFF,YELLOW,SM]
   ONEY TRANSFER[RVSON,C1,SPC13]"
220 GOSUB 7000
221 S=2:GOSUB 112:PRINT"[ST]O CATEGORY :-";:INPUT CC$
222 IF CC$=""THEN 114
223 PP=ASC(CC$)-64:GOSUB 262
224 PRINT"[CLEAR,C1,RVSON,SPC13,RVSOFF,YELLOW,SM]
   ONEY TRANSFER[RVSON,C1,SPC13]"
225 GOSUB 7000
226 C=ASC(C$)-64:CC=ASC(CC$)-64
227 PRINT"[DOWN,WHITE,ST]RANSFER FROM :-"
   :PRINT"[RVSON]"A$(C) "[RVSOFF,SPC]TO[SPC,RVSON]"A$(CC)
228 LK=BA(C,RO%(C)-1):GOSUB 264
229 PRINT"[DOWN2,C7,SM]ONEY IN ACCOUNT :-£"BA$
230 PRINT"[DOWN2,CYAN,SA]MOUNT :-":INPUT WI(C,RO%(C))
231 IF WI(C,RO%(C))=<VAL(BA$)THEN 234
232 PRINT"[CLEAR,DOWN10,YELLOW,RVSON,RIGHT11,SI,SN,SS,SU,
   SF2,SI,SC,SI,SE,SN,ST] [SF,SU,SN,SD,SS]"
233 FOR T=1 TO 1500:NEXT:GOTO 114
234 INPUT"[DOWN2,GREEN,SD]ETAILS :-";DA$(C,RO%(C))
235 DE(C,RO%(C))=0:BA(C,RO%(C))=BA(C,RO%(C)-1)-WI(C,
   RO%(C))
236 BA(CC,RO%(CC))=BA(CC,RO%(CC)-1)+WI(C,RO%(C))
   :DE(CC,RO%(CC))=WI(C,RO%(C))
237 WI(CC,RO%(CC))=0
238 XX$(C,RO%(C))="*"+CHR$(CC+64):XX$(CC,
   RO%(CC))=CHR$(C+64)+"*"
239 DA$(CC,RO%(CC))=DA$(C,RO%(C))
240 X=C:IF RO%(X)=49 THEN GOSUB 63
241 X=CC:IF RO%(X)=49 THEN GOSUB 63
242 IF RO%(C)<49 THEN RO%(C)=RO%(C)+1
243 IF RO%(CC)<49 THEN RO%(CC)=RO%(CC)+1
244 GOTO 114
254 A=A+1:PRINT"[CLEAR,C1,RVSON,SPC13,RVSOFF,YELLOW,SA]DD
   A CATEGORY[RVSON,C1,SPC13]"
255 GOSUB 7000
256 IF A>20 THEN PRINT"[DOWN3,WHITE]"TAB(15)    "[RVSON,
   SL,SE,SD,SG,SE,SR,SPC,SF,SU,SL2,RVSOFF]"
   :FOR T=1 TO 1500:NEXT:A=A-1:GOTO 43
257 PRINT"[DOWN2,C6]ENTER NEW CATEGORY "CHR$(A+64)
   :INPUT"[C7,DOWN]";A$(A)
258 RO%(A)=1:BA(A,RO%(A))=0:GOTO 114
259 RUN
260 IF PO<2 OR PO>A THEN PRINT"[CLEAR]":GOTO 211
261 RETURN
262 IF PP<2 OR PP>A THEN PRINT"[CLEAR]":GOTO 219
263 RETURN
264 DE$=STR$(DE(X,B))
265 IF DE(X,B)=INT(DE(X,B))THEN DE$=DE$+".00":GOTO 268
266 IF RIGHT$(DE$,1)<>"0"AND MID$(DE$,LEN(DE$)-1,
   1)<>"."THEN 268
267 DE$=DE$+"0"
268 DE$=MID$(DE$,2):WI$=STR$(WI(X,B))
269 IF WI(X,B)=INT(WI(X,B))THEN WI$=WI$+".00":GOTO 272
270 IF RIGHT$(WI$,1)<>"0"AND MID$(WI$,LEN(WI$)-1,
   1)<>"."THEN 272
271 WI$=WI$+"0"
272 KL=LK:LK=FNR(KL)
273 WI$=MID$(WI$,2):BA$=STR$(LK)
274 IF LK=INT(LK)THEN BA$=BA$+".00":GOTO 277
275 IF RIGHT$(BA$,1)<>"0"AND MID$(BA$,LEN(BA$)-1,
   1)<>"."THEN 277
276 BA$=BA$+"0"
277 BA$=MID$(BA$,2):RETURN
278 GET C$:IF C$=""THEN 278
279 IF C$=CHR$(13)THEN RETURN
280 AC$=AC$+C$:PRINT"*";:GOTO 278
281 IF LEN(DE$)=4 THEN P1=P1+5
282 IF LEN(DE$)=5 THEN P1=P1+4
283 IF LEN(DE$)=6 THEN P1=P1+3
284 IF LEN(DE$)=7 THEN P1=P1+2
285 IF LEN(DE$)=8 THEN P1=P1+1
286 IF LEN(WI$)=4 THEN P2=P2+5
287 IF LEN(WI$)=5 THEN P2=P2+4
288 IF LEN(WI$)=6 THEN P2=P2+3
289 IF LEN(WI$)=7 THEN P2=P2+2
290 IF LEN(WI$)=8 THEN P2=P2+1
291 IF LEN(BA$)=4 THEN P3=P3+6
292 IF LEN(BA$)=5 THEN P3=P3+5
293 IF LEN(BA$)=6 THEN P3=P3+4
294 IF LEN(BA$)=7 THEN P3=P3+3
295 IF LEN(BA$)=8 THEN P3=P3+2
296 IF LEN(BA$)=9 THEN P3=P3+1
297 IF LEN(DE$)=3 THEN P1=P1+6
298 IF LEN(WI$)=3 THEN P2=P2+6
299 IF LEN(BA$)=3 THEN P3=P3+7
301 RETURN
302 IF LEN(DE$)=5 THEN P$="[SPC4]"
303 IF LEN(DE$)=6 THEN P$="[SPC3]"
304 IF LEN(DE$)=7 THEN P$="[SPC2]"
305 IF LEN(DE$)=8 THEN P$=" "
306 IF LEN(WI$)=4 THEN Q$="[SPC5]"
```

```
307 IF LEN(WI$)=5 THEN Q$="[SPC4]"
308 IF LEN(WI$)=6 THEN Q$="[SPC3]"
309 IF LEN(WI$)=7 THEN Q$="[SPC2]"
310 IF LEN(WI$)=8 THEN Q$=" "
311 IF LEN(BA$)=4 THEN R$="[SPC6]"
312 IF LEN(BA$)=5 THEN R$="[SPC5]"
313 IF LEN(BA$)=6 THEN R$="[SPC4]"
314 IF LEN(BA$)=7 THEN R$="[SPC3]"
315 IF LEN(BA$)=8 THEN R$="[SPC2]"
316 IF LEN(BA$)=9 THEN R$=" "
317 IF LEN(DE$)=4 THEN P$="[SPC5]"
318 IF LEN(WI$)=3 THEN Q$="[SPC6]"
319 IF LEN(BA$)=3 THEN R$="[SPC7]"
320 GOTO 78
321 PA$=" AAAAAAAAA":PE$="[SPC30]AAA"
322 PB$="[SPC11]99999.99"
323 PC$="[SPC21]99999.99"
324 PD$="[SPC31]99999.99"
325 PF$="[SPC39]A"
326 OPEN 3,4,1:OPEN 6,4,6:OPEN 10,4,10
327 OPEN 2,4,2
328 OPEN 1,4,7:PRINT#1,CHR$(14)
329 PRINT#6,CHR$(0):RETURN
330 PRINT"[BLACK,CLEAR]"
331 INPUT"[BLACK,CLEAR]";C1
332 :
333 PRINT"[DOWN,C3,RVSON,SC,SH,SO2,SS,SE]
    :-[RVSOFF,BLACK]"
334 PRINT"1 =[SPC,SA]DD":PRINT"2 =[SPC,SS]UBTRACT"
    :PRINT"3 =[SPC,SM]ULTIPLY":PRINT"4 =[SPC,SD]IVIDE"
    :PRINT"5 =[SPC,SA]NSWER[DOWN2]"
335 GET C$:IF C$=""THEN 335
336 IF C$<>"5"THEN INPUT C2
337 ON VAL(C$)GOTO 338,339,340,341,342
338 C1=C1+C2:GOTO 332
339 C1=C1-C2:GOTO 332
340 C1=C1*C2:GOTO 332
341 C1=C1/C2:GOTO 332
342 LK=C1:GOSUB 264:PRINT"[C3,RVSON,SA,SN,SS,SW,SE,SR,SPC]
    :-[BLACK,RVSOFF]▒"BA$
343 PRINT"[DOWN,SPC8,SF]1 =[SPC,SM,SO,SR,SE,SPC5,SF]3 =
    [SPC,SR,SE,ST,SU,SR,SN]"
344 GET C$:IF C$=""THEN 344
345 IF C$="[F1]"THEN 331
346 IF C$="[F3]"THEN GOTO 114
350 IF A1$="1"THEN A2=1:A3=16
351 IF A1$="2"THEN A2=17:A3=32
352 IF A1$="3"THEN A2=33:A3=48
353 RETURN
355 IF RO%(X)<18 THEN A1=1:RETURN
356 IF RO%(X)<33 THEN A1=2:RETURN
357 IF RO%(X)>32 THEN A1=3:RETURN
7000 PRINT"[UP,RED,CT40]":RETURN
```

■

Club 128



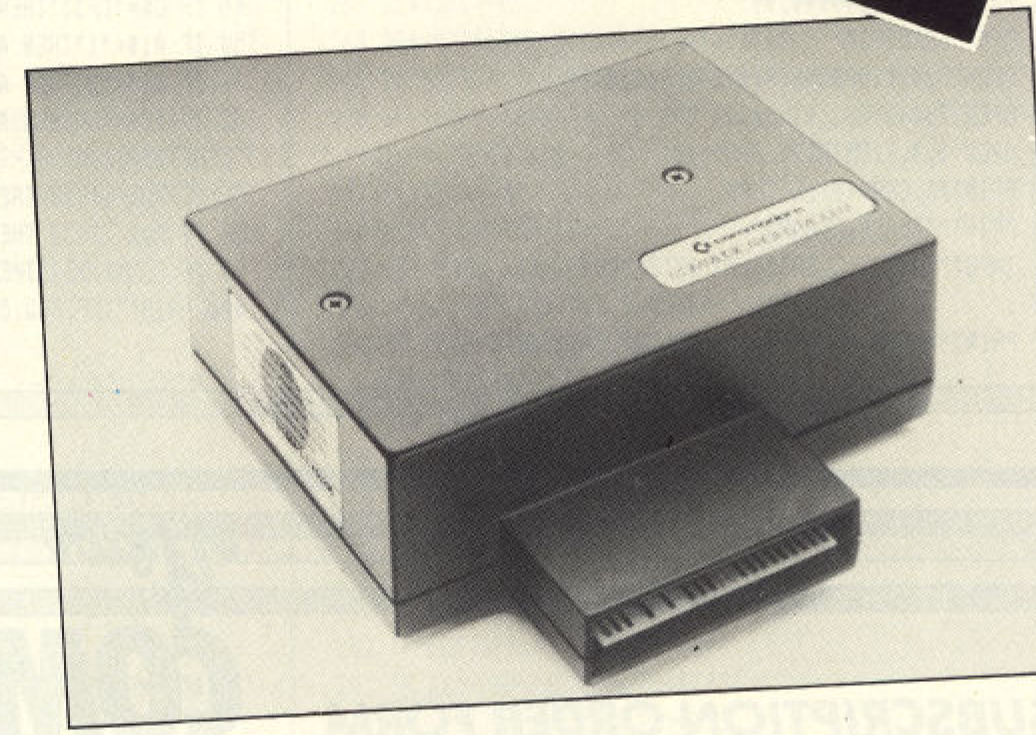
HAVE YOU EVER WANTED TO GET INTO communications? Would you like to get in touch with other Commodore owners? Well, Your Commodore, together with Compunet, are pleased to announce the start of Club 128, and give you a chance of a special membership offer.

Club 128 will appear on Compunet as a 'free access' area open to all CNET subscribers.

The club will be the focal point for all 'serious' users of Commodore computers, hence the '128' affiliation. The 128 doesn't mean that owners of C64s can't join. In fact the club is open to all Compunet members though only 'serious' users are expected to make a great deal of use of it.

Many sections of Compunet are due to be 'pulled in' to this online area, including the 'Briefcase' business section that is already in existence.

Your Commodore will have a magazine section within the club area where you will be able to find news and the latest programs that are published in the Magazine.



The Offer

So what do you get and what does it cost?

Your Commodore is offering a special membership to the Club for £69.99 (inc VAT and P&P in the UK) including:

A Commodore 64/C128 Communications modem (worth £79.99)

A three month GOLD subscription to Compunet (worth £15.00)

A six month subscription to Your Commodore

*** PLUS ***

£10.00 of discounts on selected business and communications software, including software for accessing Viewdata and Electronic Mail services such as Prestel and Telecom Gold.

In order to take advantage of this special offer you simply need to send a cheque or Postal Order made payable to Club 128 to the address on the below order form.

I enclose a cheque made payable to Club 128 for £69.99 for my Gold membership to Compunet, Commodore modem and six month subscription to Your Commodore.

Name ....

Address ....

....

....

Postcode ....

Please allow at least 28 days for delivery. Order should be sent to: Club 128 Compunet Teleservices Ltd, 7-11 Minerva Rd, London NW10 6HJ.

YOUR COMMODORE

Lineage: 40p per word.
(46p incl. VAT).
Semi display: £9.00 per single column centimetre (plus VAT). Ring for information on series bookings/discounts.

All advertisements in this section must be prepaid.
Advertisements are accepted subject to the terms and conditions printed on the advertisement rate card (available on request).

01-437 0699
EXT. 291

Send your requirements to:
STUART TAYLOR
ASP LTD, 1 GOLDEN SQUARE, LONDON W1

SOFTWARE

SOCCER MANAGER — Manage your own team, 4 divisions, F.A. Cup, transfers tactics etc.
RACENIGHT — Run your own racenight, up to 10 players. Exciting horse racing game.
SOCCER QUIZ — Test your soccer knowledge, hundreds of questions.
Any one for **£3.99,** 2 for **£6.99,** 3 for **£8.50**
T. PALMER, 59, Old Park Road, Sheffield 8. Tel: 0742 749498.

SOUTH EAST KENT SOFTWARE
Dept Y/C P.O. Box 152 Orpington, Kent
We stock for Spectrum, Amstrad, Commodore. Send S.A.E. for full list and prices. Please state which. Postage/Packing overseas £1.00.

COMMODORE 64 GAMES (TAPE)

	RRP	OUR PRICE
Commando	£9.95	£7.31
Beach Head II	£9.95	£6.75
Jeff Capes	£8.95	£6.65
Summer Games II	£9.95	£6.71
Transformers	£8.95	£6.65
Winter Games	£9.95	£7.31
Never Ending Story	£9.95	£6.51
Quick Shot II Joystick	£11.95	£7.95

And Many More Titles
Prices Now **Include** Postage/Packing

BUSINESS (DISK)

	RRP	OUR PRICE
Superscript	£69.95	£52.95
Superbase 64	£99.95	£68.00
Home Accounts	£24.95	£19.95

EDUCATION
English, Maths, German, French, History
RRP £9.95 **OUR PRICE £7.95**
Cheques/PO's payable to: **J. Ferguson.**
Please allow 21 days delivery

ALPHASOFT
CBM64 Software - 25% Reductions

Kung-Fu Master **£7.45**	Comic Bakery **£6.90**
Desert Fox **£7.45**	Yie Ar Kung Fu **£6.90**
Spy Vs Spy II **£7.45**	Uridium **£6.90**
The Eidolon **£7.45**	Ball Blazer **£7.50**
Merlenary **£7.45**	Bounder **£7.45**

AND MANY MORE!
Cheque/P.O. to **ALPHASOFT**
115 Parsonage Road, Withington, Manchester M20 9WZ

BANKER 64

If you are fed up not being able to keep track of your bank account this program may be just what you are looking for. Simple to use as it is menu driven it will hold full details of Credits, Debits, Standing orders and Direct Debits and will automatically place them in date order. It will give you statements to screen or printer including the 1520 plotter printer. Records can be stored on both disk and tape for complete back-up. Entry to any account is by name and Security code which can be changed as necessary. This m/c program is very good value at **£6.25 disk, £4.00 tape.**

RAMFILE

Keep track of your Software, Records, Stamp Collection, Addresses etc. with this easy to use menu driven filling system. Entry titles are user defineable. Powerful options include search and amend routines. Print routines allows printing of entry title, selected entry or complete file to any printer including the 1520 plotter. Files can be stored on both disk and tape.
This m/c program is very good value at **£6.25 disk, £4.00 tape.**

***Special Offer* — Both programs £10.Disk Both programs £7.00 Tape**
All prices include VAT P&P Cheques & P.O. to
RJP ELECTRONICS
27 Denham Way, Camber, Rye, East Sussex TN31 7XP Tel: (0797) 225549

DERBY DAY - C16 Can you pick a winner in this Horse-Racing Spectacular? On Screen Action, visual form guide. Over 50 horses, thousands of race combinations. Realistic betting from different meetings. Plus many more features. Send £5.75. Enquiries welcome. Cheque P/O to Bloomfield Software, P.O. Box 69, Blackpool FY2 9EX.

THE OPEN Golf Game. Plus 4 or C16 (tram). Sound, colour graphics £1.95. Cheque/P.O. to J.C. Showell, 3 Silver Birches, Elstead, Surrey.

CLAIRES COMPUTERS
Commodore 64 special offers

DISC	OUR PRICE
Fighter Pilot	£6.50
They Sold A Million	£5.95

TAPE	OUR PRICE
Pit Stop II	£4.99
Winter Games	£4.99
Theatre Europe	£4.50
Frankie Goes To Hollywood	£3.50
The Fourth Protocol	£3.50
Spitfire 40	£4.50
Spitfire Ace	£1.99
Jump Jet	£3.50
Ace	£4.50
Zaxxon	£2.50
Rock 'n' Wrestle	£4.99

This is just a Small selection from our Stock. Send now for our latest list.
Cheque/P.O.s to
CLAIRES COMPUTERS
16 Treganoon Road, Mount Ambrose, Redruth, Cornwall TR15 1QZ. Tel: 0209 211930.
FOR YOUR SOFTWARE NEEDS

SOFTWARE TO SELL!
FOR ADVERTISING DETAILS PHONE DUNCAN NEILSON ON 01-437-0626

REPAIRS

COMMODORE REPAIRS
Commodore repairs by Commodore approved engineers. Repair prices: Vic 20 modulator **£8.05**, Vic 20 **£15.00**, CBM 64 from **£20.00**, C2N from **£9.00**, 1541 from **£20.00**, C64 psu **£13.50**, Spectrum **£16.50**, Dragon **£32.00**, Electrons, Atari.
For more information SAE or telephone:
G.C. BUNCE & SON,
36 Burlington Road, Burnham, Bucks SL1 7BQ.
Tel: (06286) 61696.
9am-6pm Mon-Fri.

TRIDENT ENTERPRISES LTD
Commodore 64 — **£15.** Vic 20 — **£12.50,** Spectrum — **£10.50,** QL — **£30,** 1541 Disk Drives — **£20,** and all Commodore systems. Send faulty machine with note describing fault to:

UNIT 7, WENTWORTH INDUSTRIAL COURT, GOODWIN ROAD, BRITWELL, SLOUGH TEL: 0753 21391

Overseas enquiries welcome.
(Above prices are exclusive of parts and VAT).

GENERAL

BUY OR SELL your unwanted software, books, and hardware. We put you directly in contact with each other in your area. Details: Second Hand Computer Sales Agency. 38, Bowling Alley Walks, Dorchester, Dorset. Tel (0305) 64849 (24 hrs).

DISCS

!!! FREE DISKS !!!
For every 10 ordered we supply 11

5¼"
SS/DD **£10.99** SS/QD **£13.19**
DS/DD **£13.19** DS/QD **£15.39**

3½"
S/Side **£27.49** D/Side **£32.99**

All prices **INCLUDE VAT**
Delivery is **FREE**
All disks guaranteed for 12 months
Cash with order please to
HOLLBARN Ltd., Dept CM1, 8 Alexander Road, Stotfold, Hitchin, Herts. SG5 4NA.

"MR FLOPPY" Discs. The Universal discs will work reliably on ANY drive. D/S D/D 96tpi 80tr. 10 for £15, 25 £35, 50 £65, lifetime warranty. SantellSoft, 10 Pennine Drive, NW2 1PB, 458 1491. Unbranded discs D/S 80tr, 10 for £12. Prices inclusive.

EPROMS

PROM-64

EPROM programmer for 8K, 16K, ZIF socket, Vpp generator & software on tape	£34.75
64-CARTridge, 2 × 8K	£5.95
DEMO EPROM 8K	£4.95
DHOBI-2 EPROM eraser	£22.95

UK + 15% VAT
Cambridge Microelectronics
One Milton Road, Cambridge CB4 1UY
Tel: (0223) 314814 Tlx: 81574

HARDWARE

Super Breaker Back-Up Board

For the Commodore 64, CBM 128 and VIC-20. When a second CBM type cassette unit is connected to the board it will make security back-up copies of cassette based software, including Turbo/Fast Loaders, Basic and/or Machine Code and Data Files. No additional software is needed. Very User Friendly, the back-up copy is made on the second cassette while the original is loading on the first. Price at £8.95 Plus P&P.

RESET CARTRIDGE for CBM 64. This cartridge will reset all, yes all, memory resident programs. Including all programs where standard reset switches fail. Price £7.95 + P&P.

P&P 55p UK (Europe add £1.00, elsewhere add £2.00). Send cheque/postal order, or a stamp for details to:
C. Kent, 26 Northcape Walk, Corby, Northants. NN18 9DQ.
For enquiries telephone (0536) 742622

SPORT

Horse form CBM 64/128 graphics/sound, printer/screen output, save-load race files, easy data entry. For the more discerning flat racing man. Horse form gives winners. **£7.95** cassette.
Cheque/PO to:
SEESAW SOFTWARE,
Box No. 212, YC Argus Specialist Publications, No. 1, Golden Square, London W1R 3AB.

When replying to classified advertisements readers are recommended to take steps to protect their interest before sending money.

WARNING NOTICE

Advertisements placed in this magazine are to be in strict compliance with our standard conditions (copies of which conditions are available on request) and on the clear understanding that the advertiser warrants that his advertisement(s) does not infringe any copyright or condition of sale of any interested party in the advertised product.

Further, the advertiser indemnifies the proprietors of this magazine in respect of costs, damages, or any other claims brought against them as a result of legal action arising from the publication of the advertisement.

Any breach of these terms or the said conditions may result in prosecution of the advertiser by the proprietors.

FOR HIRE

HIRE HIRE

STAFFS COMPUTER LIBRARY

CBM 64 OWNERS

Join the friendly Library. We give you what you want. If a title is not on our list then we will get it, if it's still available. (Games only)

- All the latest Titles
- Full 7 Days Hire
- Low Hire Cost. From 75p
- Up to 4 Games per week
- All games original
- Lots of American Titles
- Low Hard and Software prices
- Membership ONLY £5.00

Free m/c Monitor on joining

Send your cheques/postal orders payable to:
STAFF COMPUTERS
The Newlands, Greensome Lane, Stafford

COMMODORE 64 cassette library. Hire latest games and educational software, only one price £1.50 per cassette 7 days hire. Details S.A.E. please: C.C.L., 19 Darrell Way, Abingdon, OX14 1HQ.

VIC-20/CBM-64 SOFTWARE HIRE

Free Membership. Hundreds of Titles to choose from. Send S.A.E. for your Free Hire Kit:
VSH (YM), PO BOX 65, NOTTINGHAM NG1

TERMS

CLASSIFIED ADVERTISING TERMS & CONDITIONS

Our terms for new advertisers (semi-display and lineage) are strictly pro-forma payments until satisfactory reference can be taken up (excluding recognised advertising agencies). Cheques and PO's should be crossed and made payable to:
ARGUS SPECIALIST PUBICATIONS LTD.,
and send together with the advertisement to:
THE CLASSIFIED DEPT., L/H, NO: 1 GOLDEN SQUARE, LONDON, W1R 3AB.
There are no reimbursements for cancellations. Advertisements arriving too late for a particlular issue will be inserted in the folowing issue unless accompanied by instructions to the contrary.
All advertising sales are subject to Government regulations concerning VAT. Advertisers are responsible for complying with the various legal requirements in force eg: The Trade Description Act, Sex Discrimination Act & The Business Advertisements (Disclosure) Order 1977.
FULL TERMS & CONDITIONS OF ADVERTISING AVAILABLE ON REQUEST

SOFTWARE APPLICATIONS

HOME ACCOUNTS

Put your house in order! Probably home computing's best use! Comprehensive coverage of bank accounts. Credit cards, HP in-built accuracy check. Records all transactions. Projects cash-flow for any period ahead. Available for Complete Commodore Range. £8.45 or *free* details from:

Discus Software
Freepost
Beach Approach
Brixham TQ5 8BR
Tel: 08045 55532

HARDWARE

★ BACK-UP-SYSTEMS FOR CBM-64 & VIC-20 ★

The BACK-UP-BOARD Mk. 1 is the fastest, and easiest way to back up your tape programs. With this board you can connect a second CBM cassette deck to your computer. You can copy a program on the second deck while the original loads from the first. You can back up all fast loaders, data files, etc. BACK-UP-BOARD Mk. 1 £10.00 inc. P&P.

BACK-UP-BOARD Mk. 2. This new version of our famous BACK-UP-BOARD is now proving so popular it is outselling our original Mk. 1. This new board now works two ways. Switched to 'NORMAL' it operates as the Mk. 1. Switched to 'GENERATE', '64' owners can use our special transfer software to regenerate a copy of a program on the second deck as it loads from the first. This method can produce a better quality copy than the original. BACK-UP-BOARD Mk. 2 & Software £14.50 inc. P&P. Conversion service. Return your Mk. 1 Board, if purchased from us, to be modified to Mk. 2 for £4.50 including software & P&P.

AZIMUTH HEAD ALIGNMENT TAPE by INTERCEPTOR for '64'. Use this test tape with your '64' to adjust the heads of your decks to make a perfectly matched pair. Full instructions and screwdriver supplied. £8.50 inc. P&P.

ROM SWITCH/RESET BOARD. Fit this board in the expansion port of your '64' and you can reset ALL programs. Resets programs that a standard reset switch can't. A socket for a 2764 8K eprom is also fitted. Full instructions supplied. £8.95 inc. P&P.

Please send SAE for more details of the above products. Send your cheque/PO to:

TURBOTRONIC
46 Ripon St., Parkinson Lane, Halifax, West Yorkshire HX1 3UG.
Tel: (0422) 52020
(Europe add 50p outside Europe add £1.50)

FOR A POSITIVE RESPONSE! PHONE FOR A COMPREHENSIVE ADVERTISING PLAN ON 01-437-0626

GAMES

WORD DEALER, an exciting word game for up to four players. Four play levels too! Send cheque/P.O. £3.25 to R. Porter, Woodside House, Abbey Road, Scone, Perthshire, PH2 6LL (C16 and plus/4).

UTILITIES

The long awaited 1541 Utility Disc V1.4!!!

As we promised it features:

* Disc copier V1.2
* File copier V1.2 : copies filed up to 196 blocks long.
* Menu driven disc aid system with 12 useful commands.
* Autoboot utility.
* Program address reader
* Quality packing and documentation.

All this for just **£9.95** on disc including 1st class post packing and VAT.

Please make cheque/P.O. payable to

AURASOFT (Dept M1)
4 Ash Grove, Nelson, CS4 66LR.

COPY DEADLINES FOR YOUR COMMODORE

AUG — 2nd MAY
SEPT — 24th JUNE
OCT — 28th JULY
NOV — 26th AUG

DON'T MISS OUT PHONE 01-437-0626 FOR DETAILS

FOR SALE

MOONRAKER £7.95
CBM64 Space Game features fast
***Fly-into Rolling Radar Display**
***Infra Red Target Screen**
Altitude, Time, Score Indicators, Fuel/Purchase Option, Promotion.
Send your name, address & PO/cheque to:
J & J Gilmour Ltd.,
Dept YC486, 4 Copse Close, Liss, Hampshire GU33 7EW, England.
Tel: 0730-893430.

TO ADVERTISE IN THE ONLY INDEPENDANT COMMODORE MAGAZINE PHONE DUNCAN NEILSON NOW ON 01-437-0626

GADGET 64 £6.95
This 100% machine code CBM64 cassette utility offers: Append, Delete, Renumber (including Goto, Gosub, etc.) Satisfyingly simple to use. Send your Name, Address & PO or Cheque for £6.95 (P&P Free) to
J & J Gilmour Ltd.,
Dept YC486, 4 Copse Close, Liss, Hampshire GU33 7EW, England.
Tel: 0730-893430.


DRIVE AMERICANA
FOR MORE
MPG
MICROPOWER · PER · GAME
Americana Software, all American action at a superb value for money price of only
£2.99
COMING SOON
SLAMBALL
COMMODORE 64/128
NEW YORK CITY
COMMODORE 64/128 · ATARI
BREAKDANCE
COMMODORE 64/128
SHAMUS
COMMODORE 64/128 · ATARI
SCOOTER
COMMODORE 64/128 · ATARI
OLLIES FOLLIES
COMMODORE 64/128 · ATARI
LICENSED TO THRILL
AMERICANA
SOFTWARE
AMERICANA SOFTWARE, UNIT 10, PARKWAY INDUSTRIAL CENTRE, HENEAGE STREET, BIRMINGHAM B7 4LY. TEL: 021-359 8881